AFRL-IF-RS-TR-2003-187
Final Technical Report
July 2003

THE THEORETICAL UNDERPINNINGS OF PREDICTIVE BATTLESPACE AWARENESS (PBA)

Prediction Systems, Incorporated

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-187 has been reviewed and is approved for publication.

APPROVED: /s/
TIMOTHY E. BUSCH
Project Engineer

FOR THE DIRECTOR:
/s/
JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
JULY 2003

3. REPORT TYPE AND DATES COVERED
Final Sep 02 – May 03

4. TITLE AND SUBTITLE
THE THEORETICAL UNDERPINNINGS OF PREDICTIVE BATTLESPACE AWARENESS (PBA)

5. FUNDING NUMBERS
C - F30602-02-C-0230
PE - 62702F
PR - 459S
TA - BA
WU - 22

6. AUTHOR(S)
William C. Cave

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Prediction Systems, Incorporated
Prediction & Control Systems Engineers
309 Morris Avenue
Spring Lake New Jersey 07762

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFSB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2003-187

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Timothy E. Busch/IFSB/(315) 330-1486/ Timothy.Busch@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
The work described here is aimed at the theoretical underpinnings of predictive battlespace awareness. The pace of combat has changed, leading to the need for continuous dynamic assessment of the situation, and prediction of future outcomes that depend upon the reactions of an intelligent adversary as well as all other factors that contribute to uncertainty. This environment has set new requirements for tools to support analysis and decision makers. One of the critical tool requirements is a new approach to simulation coupled with a capability for prediction. Simulation and prediction in a live combat environment present significant new challenges. The effort described here focuses on these challenges, treating the military planning problem as an optimal control problem. The motivation for producing a sound theoretical basis of formulation is to understand the nature of the problem to be solved, and support better judgments regarding realistic approaches to solution. It is aimed at developing a mathematical foundation for planning tools that more accurately depict the future unfolding of a dynamic battlespace.

14. SUBJECT TERMS
Predictive Battlespace Awareness, Simulation and prediction, Optimal Control Problem

15. NUMBER OF PAGES
90

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

List of Figures

PREFACE

PSI has been fortunate to work for many years with the U.S. military. The company has enjoyed the time it has been afforded to solve extremely difficult design problems in communications and control. But the real reason PSI is fortunate is the opportunity to work for military people who must deal with the harsh realities of war.

Military people are trained to think differently, especially those who rise to higher grade levels, whether enlisted or officer. This is because their major concern in life is survival. Even those in the research and development community are driven to develop systems to improve their probability of survival.

When one is concerned with survival, one is concerned with seeking the truth. One's own life may depend on it. In the military, this is emphasized by the term *ground truth*. Ground truth is the concept of what really exists. The word *concept* is used because we may or may not be able to measure it.

If we write something in a document, and that document is our ground truth, then anyone who reads it is able to *see* the ground truth - first hand. But not everyone who reads it will derive the same meaning. They may derive different perceptions of ground truth from that implied by the author. If the document gets lost, and we have to derive the meaning from memory, or by second or third hand word-of-mouth, then it becomes more likely that the original meaning can be misinterpreted - or lost. So, for the most part, we must deal with perceptions of ground truth.

Since the earliest civilizations, scientists have been engaged in developing mechanisms to seek ground truth. Looking glasses, field glasses, microscopes, and binoculars were all designed to help us forge a better picture of ground truth, often from a distance. Satellite imagery has become a significant technology for capturing ground truth.

There have been many articles written by well known authors on the subject addressed here. Various authors see the picture differently. What is popular as being the *real* picture at one time may be set aside for what's considered a better picture at a later time. If we are all seeking ground truth, then we should converge on the picture, even if our looking glasses are different.

From that standpoint, the work presented here is considered more of a looking glass than a new discovery of ground truth. Most importantly, it is aimed at achieving surprise, speed and synchronization of missions to maximize their intended effects while constraining losses.

1. BACKGROUND

This work is aimed at the Theoretical Underpinnings Of Predictive Battlespace Awareness. This area was the topic of a white paper, [1], and proposal submitted to AFRL/IFS. This document has been produced as a result of discussions with Dr. Timothy Busch, AFRL/IFS, on the contents of previous work as well as the white paper and proposal, and particularly the Dynamic Situation Awareness And Prediction (DSAP) paper authored by Alex Sisti, [2].

As the Sisti paper describes, the face and pace of combat engagements has changed. This has led to the need for continuous dynamic assessment of the situation, and prediction of future outcomes that depend upon the reactions of an intelligent adversary as well as all the other factors that contribute to uncertainty in a combat environment. Such an environment introduces requirements for new tools to support analysts and decision makers. One of the most significant tool requirements is a new approach to simulation coupled with a capability for prediction. Simulation and prediction in a live combat environment present significant new challenges. The effort described here focuses on these challenges.

2. INTRODUCTION

The Joint Force Commander (JFC) controls the coalition forces. That commander must develop the Courses Of Action (COAs) to be taken by the coalition forces. Supporting him in the development and implementation of the COAs are other national air, ground, and naval combat commanders. The Air Component Commander (ACC) is responsible for developing COAs for air operations. These COAs must take into account the potential Enemy COAs of an intelligent adversary, as well as all of the other factors that influence the unfolding outcomes of a selected COA.

The effort proposed here will focus on the U.S. part of a coalition Aerospace Operations Center (AOC), while considering the special support required from the U.S. SCIF. The number of functional areas required to support the ACC in the development of COAs and corresponding Air Tasking Orders (ATOs), and the corresponding subject area experts required to produce these plans imposes significant size and complexity requirements that must be accounted for. Each of these functional areas has specialized tools (many segregated for security purposes) to help assess the situation from their standpoint and develop subordinate plans to support proposed COAs and ATOs. This is accomplished by using various tools and approaches at the subordinate levels, and by assessing problems and likelihood of success to determine the best way to support the mission as perceived at the next superior level.

The various subordinate plans are produced with assessments of effectiveness and performance outcomes, and some measures of accuracy, confidence, and risk relative to their specific mission support areas. These are passed up an organizational hierarchy that produces higher level decisions that are passed up the line. Improving the ability to assess different COAs rapidly, and the corresponding prediction accuracy required to assess likelihood of success at any level will help the next level up.

Thus, whether we provide better tools for the coalition commander, or for his subordinates at a lower level, we will help the cause. This effort draws upon PSI's experience in supporting this process, focusing on approaches that can help at many levels, from the top down, and across the board.

3. DEFINING THE PROBLEM

Much of the effort described in this document is a result of attempting to put the DSAP problem on a sound theoretical basis. In that sense, all of the sections that follow are refinements of this section. We will start at the top and work our way through using concepts and techniques derived to solve the problems associated with designing prediction and control systems.

MILITARY PLANNING - AN OPTIMAL CONTROL PROBLEM

As described in the paper by Cave and Busch, [7], the military planning problem can be posed as an optimal control problem. To understand this, consider the following. The JFC makes decisions based upon the effects and objectives to be achieved and a large number of observable influences, including perceptions of the situation of both red and blue forces. These decisions are also based upon what the JFC perceives can be done given a review of Courses Of Action (COA) that can be taken, including the effects of the Enemy COA (ECOA). Once a COA plan is conceived, it is disbursed for implementation using the available resources.

Figure 3-1 is a simple illustration of a control system. The JFC is a key element of the control system. In this case, the control system contains a huge number of observable inputs that are percolated up from all of the intelligence and other sources that provide inputs into the decision process. Once a plan is forged, it is promulgated down to the subordinate forces.

The plan that is promulgated is not unlike the optimal control sequence put out by the controller in the classic optimal control system. Given desired objectives, the control system is constantly producing a sequence of parameters in real time that are used as controlling inputs to the system. However, in this case, the system is distributed as described in a later section.

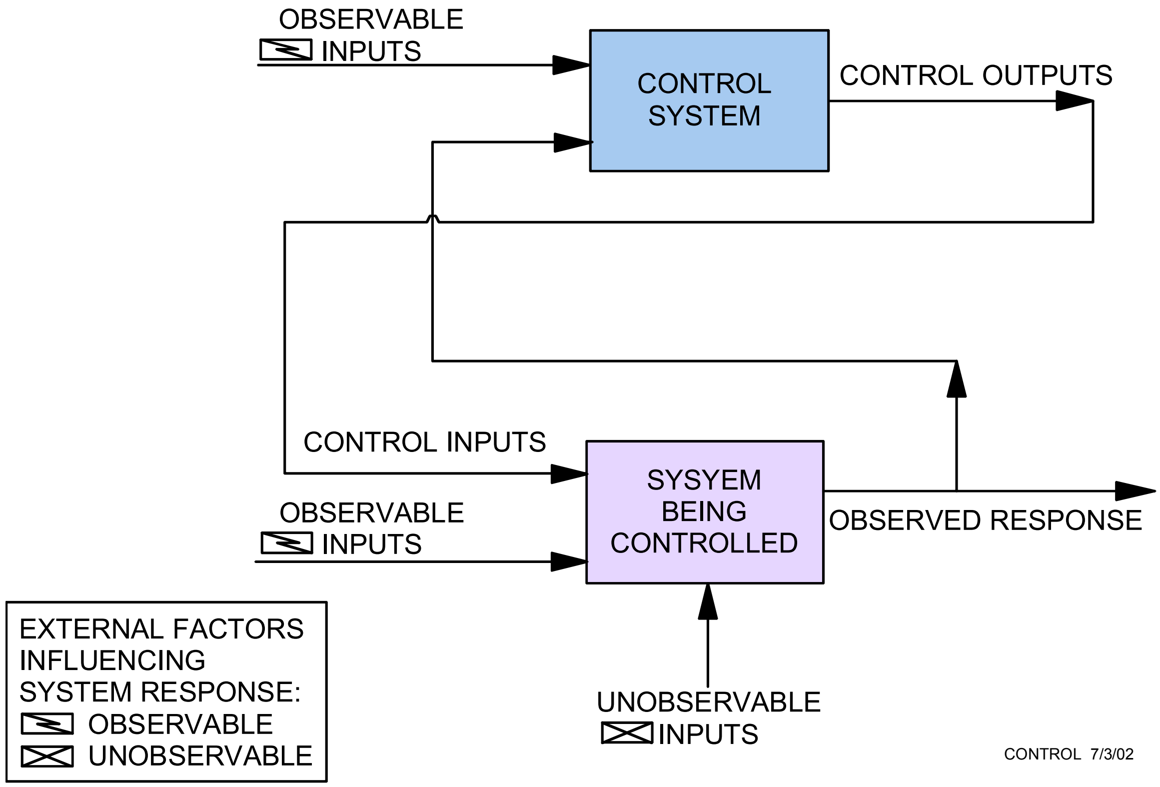

Figure 3-1. Simplified representation of a control system.

The Embedded Prediction Component Of A Control System

The sophisticated part of most control systems is the embedded prediction subsystem. This is characterized generically in Figure 3-2. The prediction subsystem takes in a selected control sequence and observable inputs up to the current time T, and produces a prediction of the resulting system response out to some desired $T+\tau$. To accomplish this, the prediction system must contain models that represent all of the complexities required to produce the predicted outcomes *with sufficient accuracy* to support the COAs. For this application, this is best accomplished using discrete event simulation and interactive graphics (a huge topic described elsewhere including many PSI documents). The control system produces sets of control sequences to the prediction system and gets back corresponding sets of predicted system responses. The optimal control problem is to come up with the control sequence that meets the *constraints* required of the system while optimizing some prescribed *objective* function. In the ensuing discussion, we will use the words *solution*, *control sequence*, and *COA* interchangeably.

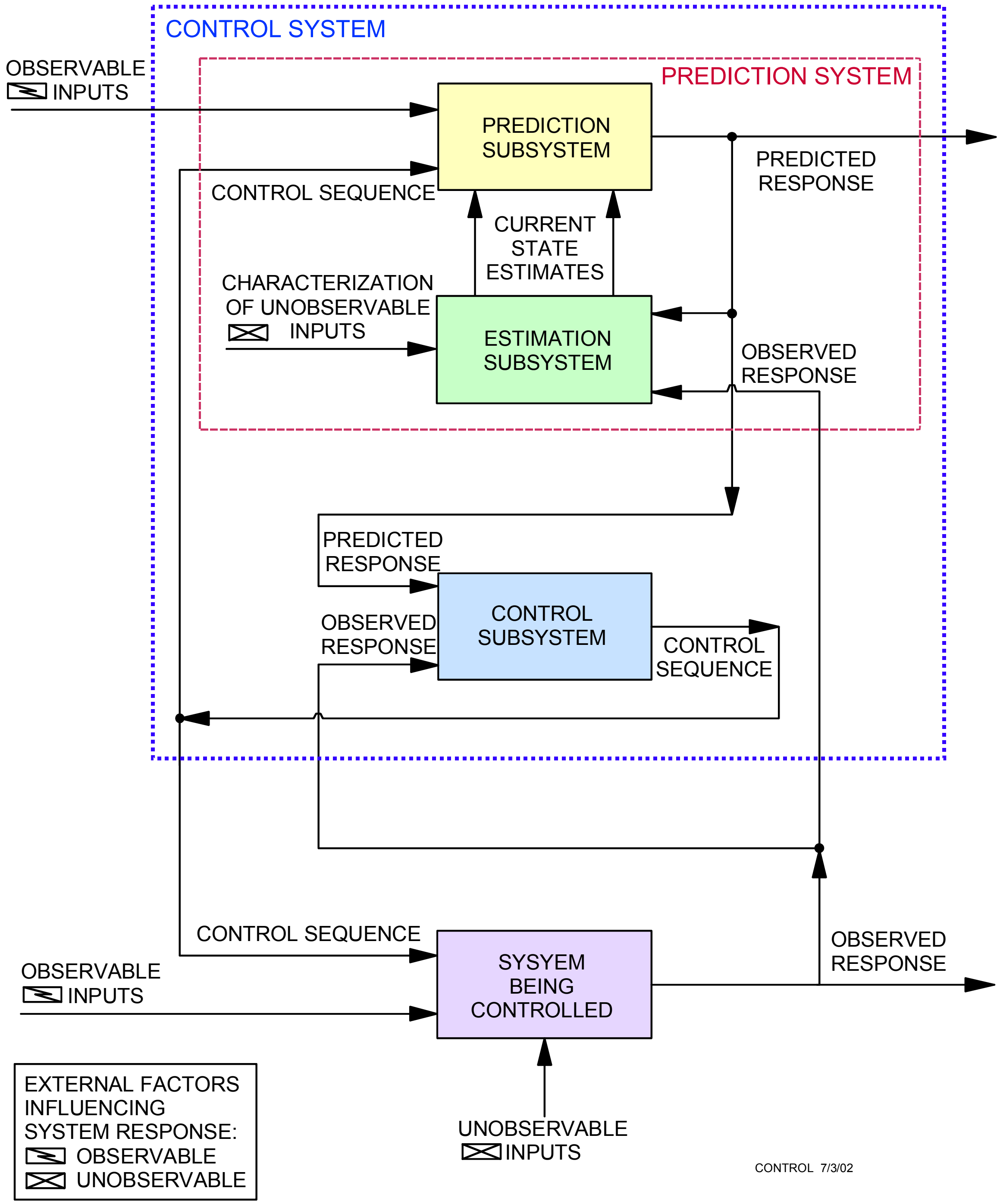

Figure 3-2. The embedded prediction component of a control system.

PREDICTION VERSUS FORECASTING

When attempting to make decisions relative to best courses of action, one wants to know what the outcomes would be for each potential course of action selected. An example is tracking the seismic behavior of a volcano, trying to determine whether or not to evacuate surrounding communities. Evacuation will cause a major disruption; but without an evacuation, many lives may be lost. When dealing with an intelligent adversary, one will be trying to postulate the adversary's actions and reactions that will affect courses of action. Predictions and forecasts are made to support the analysis and decision process that precedes courses of action.

If sufficient data and time exist, then a *prediction* can be made with the accuracy characterized. If not, one must make a *forecast*. When decisions are critical, particularly if life and death are at stake, it is important to understand the difference between prediction and forecasting to avoid misleading statements and corresponding results.

As defined here, predictions can only be made when the accuracy of the prediction mechanism can be characterized in terms of historic data used to compare *apriori* predicted outcomes to the actual outcomes. Apriori is italicized because once one has seen the outcomes, any changes to the mechanism will generally require recharacterization of the error using data that has not been seen. This point is critical and is discussed further in the next section. If one cannot perform such a characterization, then one is making a forecast.

As defined here, the difference between prediction and forecasting is independent of the prediction mechanism. One may use human instincts to make predictions. As long as the error associated with the instinctive prediction mechanism can be characterized on a consistent basis statistically, confidence levels on the error can be produced. On the other hand, one can use large quantities of historic data to feed a sophisticated mathematical model that generates future outcomes without characterizing the error. This is a forecast.

If there is no history data, one cannot characterize prediction error, and therefore one must make forecasts. This is true when new problems are being addressed that may not fit the existing prediction mechanism. In these cases, one must determine whether the changing situation still fits the prediction mechanism, or whether it is time to drop the error characterization and confidence statements and go with a forecast. This determination is relatively easy to do when the prediction mechanism is a mathematical model driven by mechanically quantified measured data. This becomes difficult when characterizing prediction error based upon human instinct.

In Section 9, we will investigate a method for combining forecasts and predictions to produce a prediction. This method will rely on the characterization of worst case outcomes, i.e., outcomes that occur based upon worst case conditions. In effect, we will condition probability statements using worst cases.

THE PREDICTION PROBLEM

Prediction of future outcomes of systems must be couched in terms of probability statements. In fact, they are conditional probability statements. For example, one can predict whether or not it will rain tomorrow in Point Pleasant, New Jersey as follows:

> ***First Prediction:*** Given the historic data on rainy days in Point Pleasant for the past 20 years, one can predict the probability of rain by dividing the number of rainy days by the total number of days. If the number of rainy days for the past 20 years (7300 days) was 730, then the probability of rain tomorrow (or any day for that matter) is 10%.
>
> ***Second Prediction:*** Given the number of rainy days in each month in Point Pleasant for the past 20 years, one can make separate predictions of the probability of rain for each month. For example, one might say that - if the month is July, then the probability of rain is 5%; - if the month is November, then the probability of rain is 15%.
>
> ***Third Prediction:*** Using knowledge of the weather patterns around the coast of New Jersey, one can rely upon the fronts moving from west to east. Given knowledge about a rainstorm heading toward Point Pleasant from Pennsylvania, one can predict the probability of rain over the next 24 hours in 6 hour increments. For example, one might say that the probability of rain is less than 2% over the next 6 hours. It is 25% for the following 6 hours. It continues to rise to 95% in the period 13 to 18 hours from now, and then falls off to 65% in 19 to 24 hours.

All of these predictions contain valid probability statements based upon historic measurements. However, the accuracy of each is obviously different. The difference in accuracy is determined by the conditioning of the probability statement. The first prediction is conditioned only upon the number of rainy days in a year, with no additional information. The second prediction is conditioned upon additional information, i.e., the number of rainy days in each month of the year. It will be a more accurate statement. The third prediction is conditioned upon a dynamic model of weather patterns. This model contains much more information than the other two, and is much more accurate.

Predictions are statements of probability of the outcome of a future event. In general, they are conditional probability statements, i.e., they are conditioned upon the information used to compute the probability. The more information one can use to condition the probability statement, the more accurate the prediction.

Human Judgment Versus Automation

The general prediction problem is simply to produce the most accurate prediction possible given the time frame and resources at ones disposal. As indicated above, predictions are probability statements that are conditioned on all of the information one can muster. Mathematical formulation is not as important as having additional information.

The classic example is that of the salesman who knows little about mathematics and uses a computer only to build a spread sheet to organize his forecasts of sales volumes of product lines for the next quarter. The numbers come from his head. The marketing department gets independent sales forecasts from a PhD statistician who uses various sophisticated statistical approaches and historic data to forecast the same sales volumes. Why does the salesman consistently come up with a much more accurate forecast? He has more information about what's going on in the market!

As indicated above, predictions are conditioned probability statements. Modelers that incorporate more information into their model will produce more accurate predictions. This information need not be in the form of historic data. It is likely that the most important information is knowledge about the structure of the system. That's why the salesman does better. He knows what is happening in the market (his system). If he's good, he has intelligence on what's changing. Are some new stores opening in two months that will be buying? Are some existing clients about to shut down? He has a more accurate model in his head than the statistician who is manipulating historic data with time-series models.

This does not imply that we cannot build a model on the computer that incorporates the salesman's knowledge. In fact, we can generate probability statements conditioned on that knowledge. If we had 100 territories each with a salesman, we could build one model with 100 instances and get them to enter their knowledge and then roll up the results - automatically. Can we get them to cooperate? Yes, if we can improve their accuracy and still make it easy for them to enter their knowledge. These are the practical problems we must deal with, and the questions we must answer.

Human Judgment And Automation In PBA

When a commander goes into the field against a new enemy, he may not have much history to go on. However, as each side makes a move, intelligence is gathered that is turned into information to condition predictions. This process has evolved over many years. Like seasoned salesmen, the participants in the planning process have been trained to look for critical pieces of information. They are also in a position to account for observable changes in the structure of the systems they are dealing with, both friendly and from the opposing force.

To not capitalize upon this knowledge would be like the statistician who competes with the salesman, using only esoteric mathematics and historic time-series data. If we want to maximize the accuracy of our predictions, we must maximize the information used to condition our probability statements. To do this, we must understand and assimilate the current planning process - in detail! To better represent these concepts, we need some additional definitions.

Defining Prediction Accuracy

Point predictions, e.g., "The probability that the temperature will be 80° tomorrow is 65%," are generally useless. To produce accurate predictions as described in [6], one must state the probability of being within specified limits as defined by a distribution. Even so, without knowledge of the accuracy of the distribution, the usefulness of the prediction is questionable. On must add a confidence statement about the distribution from which the statement was derived. For example, we predict the temperature will be between 78° and 82° tomorrow with a probability of 80%, and a confidence level of 95%. This is explained below using an example.

Consider that we build a system to predict weekly purchases of goods up to 12 weeks out. Figure 3-3 shows the historic data as well as the 12 week ahead predictions. The prediction is given in terms of a high and low value around the maximum likelihood value. The prediction is given in terms of an 80% envelope, i.e., being in between the high and low values 80% of the time. In addition, the confidence level in the prediction is 95%.

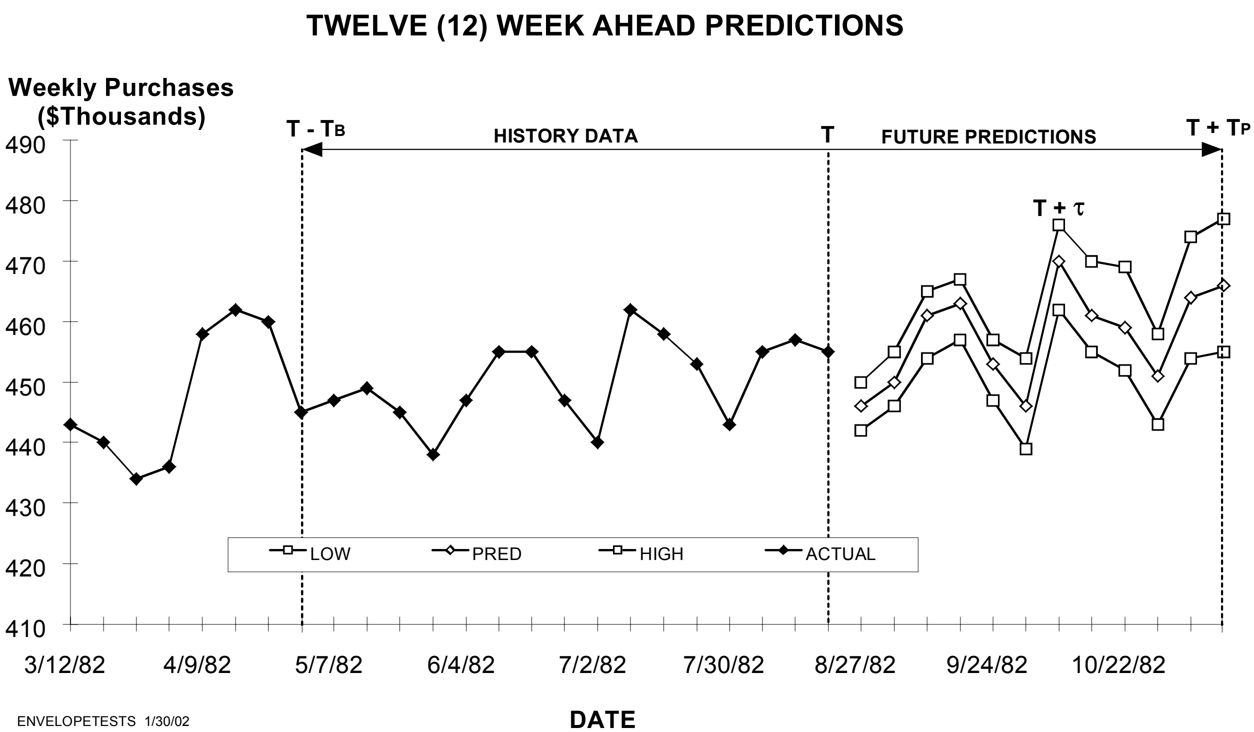

Figure 3-3. Example of a twelve step ahead prediction.

Every time a prediction is generated, the accuracy is determined by comparing the predicted value with the actual. This is done over a looking back horizon as described in [6]. In this example, the looking back horizon used is 52 weeks as shown in Figure 3-4. If 80% of the points lie within the envelope, the envelope statement is satisfied for that step. Every time we take a step, we compute this measure. The envelope statement must be correct 95% of the time.

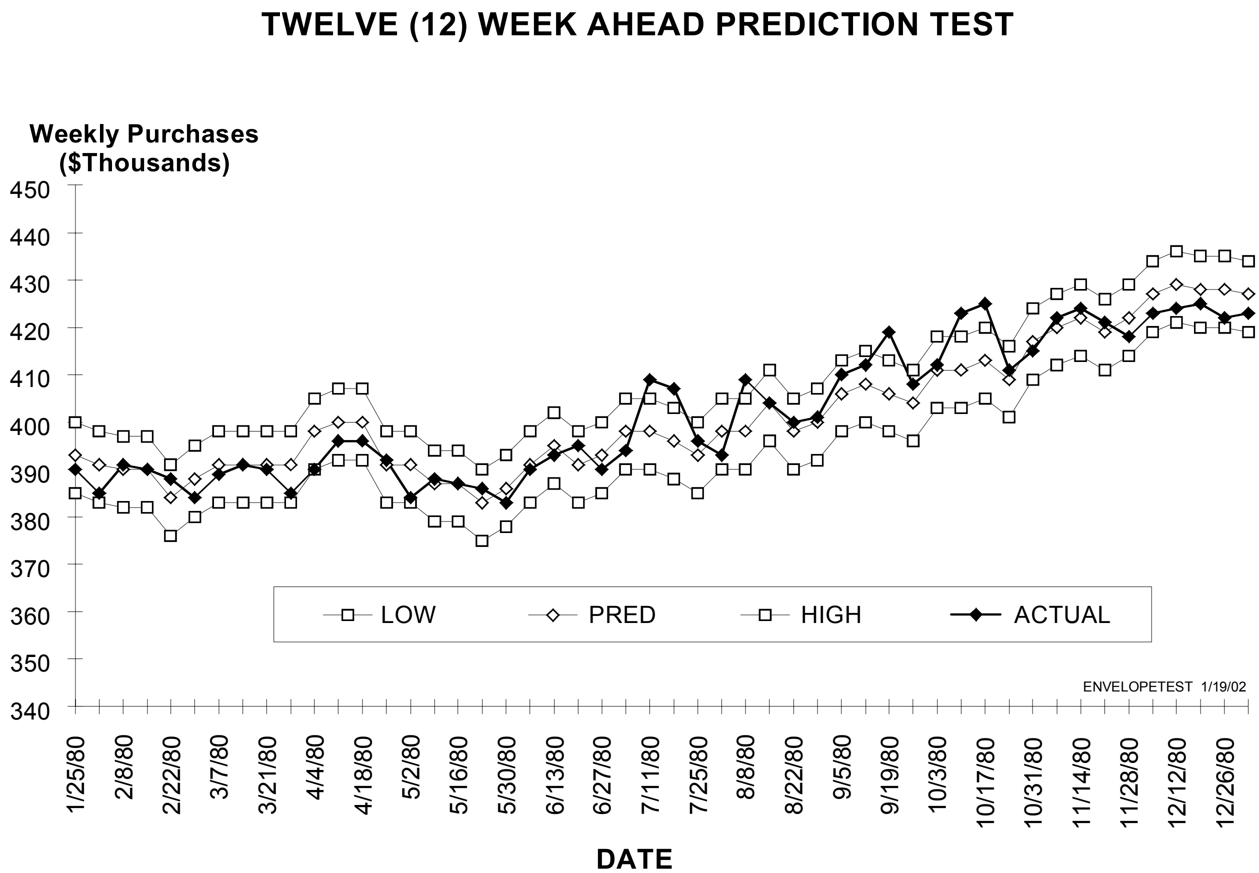

Figure 3-4. Measuring the accuracy of a twelve step ahead prediction.

To derive a probability statement, we must be referring to a distribution. The distribution may be implicit or explicit. We need not know the shape, it may be unknown. But we must have statistical data relative to the probability statement. If our predictions are tested over time, then we will be within our specified 80% limits 95% of the time. If the system we are dealing with is nonstationary (changing with time), then we may also want to specify the looking back horizon used to perform our tests.

When people in a data poor environment consider these definitions, they question the definition. As an example, when trying to forecast market demand for a new product with no history, one must consider other options. Usually one resorts to stationary statistical approaches. These approaches assume that a population exists, as do all standard statistical tests. This is not wrong. It is simply not a dynamic prediction as defined here. This problem is addressed in subsequent sections.

UNDERSTANDING THE BEHAVIOR OF COMPLEX SYSTEMS

Since our interest is predicting the responses of complex systems, we want to take maximum advantage of the information we have about those systems. As illustrated in the weather prediction problem described above, having sufficient observable data on the system is only part of the problem. Making best use of that data is typically a significant problem for complex systems. This problem is solved through representation of system behavior, i.e., how the system processes the data.

The representation may be very simple or very complex. What counts is the resulting accuracy of prediction. This typically depends upon the understanding by one or more people of the internal workings of the system, and the translation of that understanding into some form of representation or *model*.

Representing Complex Systems

Figure 3-5 illustrates a complex system with observable inputs and observable outputs. For the convenience of describing the behavior of this system, we will assume that inputs are on the left hand side of a box, and outputs on the right hand side of a box.

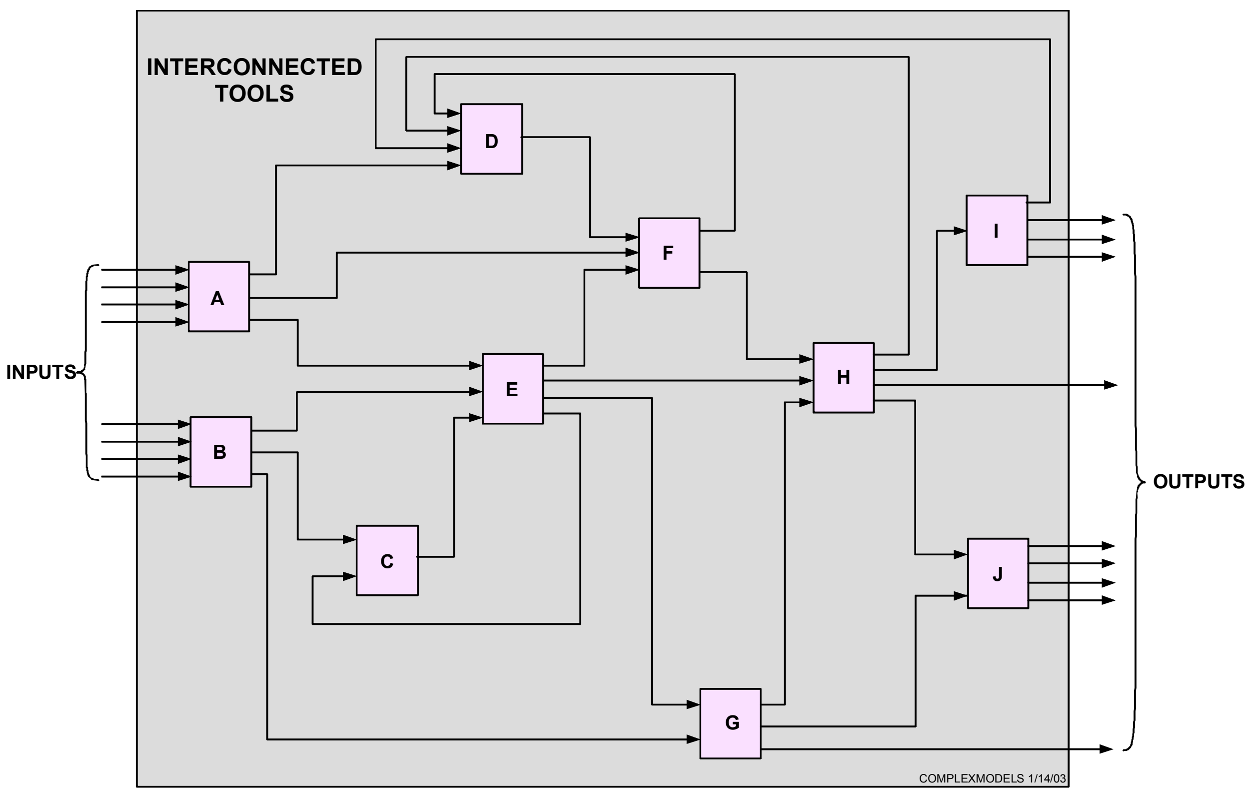

Figure 3-5. A system composed of multiple subsystems.

The system in Figure 3-5 is composed of multiple subsystems, A, B, C, etc., each of which may be complex, i.e., contain multiple sub-subsystems. If each subsystem has delays between the time an input occurs and the time a response is produced, then outputs from the overall system will be delayed by the sum of the subsystem delays encountered when starting at the input.

Note that subsystem C's output feeds into E, and that E has an output that feeds back into C. This is a feedback loop. Subsystem D participates in multiple feedback loops. Systems with feedback loops typically contain elements with nonlinear behavior to preserve stability. The combined effects of feedback and nonlinear elements typically require detailed modeling to produce accurate predictions. Assumptions based upon linearity typically do not apply, making the modeling task much more difficult.

Nonlinear - Nonstationary Systems

Approaches to solving the multi-step prediction problem have been described by many authors, see for example [3], [4], and [5]. However, these approaches depend upon algorithms that find patterns in time-series data. It has been shown by Cave, [6], that these approaches depend upon stationarity, or quaisi-stationarity in the data and can be represented by homogeneous models. In the case of nonstationary systems, e.g., those driven by nonstationary external forces, or those with internal structures that are nonstationary, approaches using homogeneous models will not provide predictions with consistent accuracy, and the methods used to measure accuracy may be questionable.

Two difficulties exist when interpreting literature describing approaches to prediction. One is the method used to test and compare prediction accuracy. The other is the use of terms, e.g., nonlinear and nonstationary. Both of these are defined in [6], and summarized here.

Prediction accuracy, or *prediction error*, can only be tested using data the modeler has not seen. If a modeler uses knowledge of "future data" to identify elements of a model, then measures of accuracy that use this same data are considered *model error*, not prediction error.

Certain types of nonlinear systems cannot be transformed into linear systems by manipulation or reordering the equations. In these cases, the test for linearity may fail, but one can manipulate the equations defining the system and produce a linear representation. We consider these *quaisi-linear* cases to be linear in nature. Similarly, nonstationary systems can be transformed into stationary systems. Again, we consider these *quaisi-stationary* cases to be stationary in nature.

4. PERTINENT CONSIDERATIONS

Future survival depends upon the speed with which one can deal with increasing complexity.

THE IMPACT OF SPEED AND COMPLEXITY ON SURVIVAL

The things we take for granted today would have boggled the minds of people just 100 years ago. Looking back 1000 or 10,000 years is awesome. Which way would any of us prefer to live? Who is better prepared to survive? The answer to the first question is generally obvious. The answer to the second requires more consideration.

Grade school children use radios, TVs, CDs, calculators, and computers. They know the difference between a tape and a DVD. But looking at the total population, how many of us understand how these complex pieces of equipment really work? Of those, how many would know how to design or build any of these items?

How many people know how to make a fire without matches? What percent of the population today could survive if stranded alone on an island? These answers will likely depend upon where we take our samples, e.g., from New York City versus the jungles of Vietnam or the mountains of Afghanistan.

Survival is the number one issue here. The U.S. is learning that there are many faces of survival. The days of firearm versus bow and arrow are long past. A high speed aircraft with smart missiles may not help preserve our own infrastructure when attacked by terrorists. The approach to survival is taking on a different meaning than historic war. The enemy situation is becoming much more complex. Accurately predicting what an adversary may do depends upon how much time he has to think, communicate, and take action. It is time to redefine the problem in light of this increasing need to deal with speed and complexity as we endeavor to survive.

Dealing With Increasing Complexity

Anyone who is familiar with the history of mathematics knows the motivations that led to the progression of numbers. It started with "whole numbers" or *integers*, and progressed to *signed integers,* then to *fractions* and *rational numbers*. This progression continued to *real numbers*, and then to *imaginary* and *complex numbers*. Each step took us into a more complex realm - not by pure imagination, but by necessity. An appropriate expression from engineering is "Necessity is the mother of invention".

But there is more to this progression than just the increase in complexity. Each of these extensions to mathematics is still referred to as a number. And each encompasses the prior. A simple whole number is a subset of a complex number. More importantly, many of the laws and transformations still apply as we move up the scale of complexity. Their interpretations are simply extended to be more general. This helps us to deal more easily with great leaps in complexity.

Selecting The Most Convenient Coordinate System

As we continue to move up the food chain of numbers and mathematics, we can group numbers into *vectors*. The position of a body in space can be described by three numbers depending upon the coordinate system we choose. And we learn in higher levels of mathematics and physics, particularly in electro-magnetic theory and partial differential equations, that problems can be solved more easily if we select the right coordinate system. For example, when a particle moves in a spherical orbit, it is much easier to describe its motion in spherical coordinates. Cartesian coordinates will work, but it takes longer to solve the problem. As a student doing homework or taking a test, one is looking for ways to beat the clock. Students who know how to apply these principles get homework done much faster. More importantly, they get answers to test problems that their competition may not finish.

Selection of the most convenient coordinate system is typically taught under the topic of *separation of variables*. One learns that the separation principle can be used if the variables form a linearly *independent* set. The property of independence can be verified using specified tests. The concept of choosing the best coordinate system and the property of independence are the important principles one can apply when dealing with complexity in a constrained time environment. We will make use of these concepts.

Einstein introduced the use of *tensors* to deal with the increasing dimensions of time, velocity, and acceleration. Control system engineers moved further to deal with *state vectors* in the *state space* framework to account for the many degrees of freedom required to characterize complex dynamic systems. The *state space* framework has been shown to be the most general representation of a dynamic system, see [19] and [23]. Providing a framework for problem description was not the only benefit of the state space approach. It also afforded faster solutions to problems that could run for days on the computers of the time.

FRAMEWORKS FOR REPRESENTING COMPLEX DYNAMIC SYSTEMS

In a competitive time-constrained environment, time (speed) is the most important factor. If two sides develop the same capability, the one that gets there first is likely to be the one that wins. When building tools to help people solve design problems or make complex planning decisions, time enters into the picture in at least two major ways.

- Development Time - the time it takes to develop the tool
- Solution Time - the time it takes to get a useful solution from the tool

One can imagine a great tool for solving a problem. But one must answer the question - can we get it built in time to accomplish our goal? Or, more importantly, will it produce valid answers fast enough if we get it built? Of course cost and risk are also major factors. However, time is usually of the essence.

Automating The Representation Process

In the early 1960's, electronic circuit designers developed automated tools for solving the complex systems of nonlinear differential equations required to represent digital waveforms in the time domain. These Computer-Aided Design (CAD) tools allowed engineers to describe large networks topologically and to write FORTRAN-like equations describing nonlinear functions. Programming skills became unnecessary. The code needed to generate and run simulations of very large networks was generated automatically. This afforded a huge leap in design productivity. It enabled the design of huge complex networks needed for integrated circuit design.

CAD system development became a business for the principals of PSI. Two systems were developed, one for continuous system modeling, e.g., for digital circuit design, and one using a discrete-time framework for the design of signal processing systems. The second used sampled data principles to reduce computation time. An underlying *state space* framework supported both products.

For large networks, the number of state variables ran into the thousands. Solving worst case design problems involved multiple optimization runs of thousands of simulations. Each simulation had to solve the optimal control problem, involving thousands of nonlinear differential equations. Speed and accuracy were the driving forces in designing these systems. If it took a computer days to get a design, only one or two test points were produced in a week - not very attractive.

Capitalizing Upon General Principals

State space was used because it provides the most convenient framework for solving any type of dynamic problem in the time domain. The general form of the solution holds for any set of independent state variables. This allowed for the development of generalized methods, e.g., optimal sparse matrix inversion and describing functions, to solve the nonlinear problem fast while ensuring algorithm convergence. The end result was to solve huge problems in minutes. However, this approach required formulating problems in a mathematical framework.

Facing Totally New Problems

In PSI products prior to 1982, models were formulated mathematically, i.e., using vectors, matrices, and systems of equations. This approach allowed the solution to be derived automatically and solved very fast. By 1982, this approach was recognized to have severe limitations when modeling communications or control systems involving algorithmic decision processes. Clients wanted to describe their problem using more general state concepts, and be able to write conditional statements within the system of equations. It was determined that these types of decision processes could be handled using the discrete event approach originally developed by Gordon in 1961, see [8] and [9].

A MORE GENERALIZED PROBLEM FORMULATION

In 1982, PSI decided to study the problem of building a discrete event simulation environment. The motivation to build a new tool was very high because of the requirement for writing decision algorithms into the models. Users wanted to break up the system of equations and embed English-like conditions and rules, e.g.,

> IF THE MESSAGE_TYPE IS CONTROL, THEN … ,
> ELSE IF MESSAGE_TYPE IS DATA, THEN … .

Additionally, complaints about the inability of existing discrete event simulation products, e.g., GPSS, SIMSCRIPT, and SLAM, to solve our client's problems. The major complaints were lack of scalability (inability to deal with increasing complexity) and excessive simulation run-times. But these complaints imposed caution. This led to an investigation of the competing product deficiencies as well as an analysis of how to formulate the basis for general solution.

At first it appeared difficult to derive a mathematical framework to support this new requirement. This caused concern about our ability to justify design decisions without a formal perspective on the problem. We appeared to be leaving the world of mathematics. Time steps were determined by the modeler in terms of scheduled events. This led to the development of a state space definition of discrete event systems. A description of this is provided in the Sections below entitled Concept Of A Generalized State Vector and State Space Definition Of A GSS Model. The differences and likenesses of mathematical and rule oriented formulations are compared in *Simulation Of Complex Systems*, [10].

Facing The Speed Issue

Because of length of competing product running times (some critically needed simulations were taking 5 to 7 days to run a 2 hour scenario), we were pushed into the desire to have the system run on a parallel machine. This was fortunate since PSI had experience in this area. Furthermore, PSI's experience in computer design, and the knowledge of how chips were evolving to support fast computing methods led to an approach that would take advantage of future hardware technology when it became available.

The need for parallel processing imposed the requirement that two or more processes would have to run concurrently if they were to run concurrently on separate processors. This implied that concurrent processes had to be independent. The property of independence implied that the processes shared no data. This led to the decision to separate data from instructions so the independence property could be tracked. Our design called for a connectivity matrix to determine what processes shared what data. Then when allocating processes to processors, the connectivity matrix would be used to determine if a process could run concurrently with those already running.

The separation of data from instructions provided very significant additional benefits. First, it allowed capitalization on the concept of independence. By limiting access to specified data structures, models could be made independent. This led to a decomposition of the simulation database into separate data structures - defined as *resources* in GSS. Instructions are grouped into sets of rules defined as *processes*. Resources and processes are grouped into elementary models. Elementary models are grouped into hierarchical models. This is illustrated in Figure 4-1, which contains a model of a local telephone system with PBXs connected to a local switch. The resources (data structures) are contained in the ovals, and the processes (instructions) are contained in the small rectangles. These can be edited directly as shown in the boxes.

In GSS, the interconnection of processes and resources is done graphically using icons and lines. This provides the ability to produce an engineering drawing of the architecture of a model, where lines connecting processes to resources determined what processes had access to which resources. Models can be connected to each other by connecting a process in one model to a resource in another. Independence of models can be visually inspected by looking at the number of lines connecting them.

The Concept Of A Generalized State Vector

Separating data from instructions clarified the meaning of the state of a model or simulation. It was defined by the state of all of the resources in that model or simulation. This led to the concept of a *generalized state vector*. One could look at the state of a simulation as one big state vector comprised of all the resources in that simulation. Alternatively, a simulation was partitioned into a set of sub-states corresponding to the resources or subvectors.

This approach allowed us to reuse many of the concepts from the state space framework. For example, the simulation state vector as used in GSS is considered to represent a *generalized coordinate system*. It is up to the modeler to come up with the best set of states to make the problem easy to solve. This implies selecting the set of resources that simplify the transformations of state that represent the dynamics of the system. These transformations are embodied in the processes. When a process runs, it starts with the initial state of the attached resources and takes them to the next state.

This is no different from the problem of picking the best set of variables or coordinate system to simplify a set of partial differential equations. As indicated above, courses that cover problem solving in the applied sciences, e.g., physics and engineering, stress that choice of a coordinate system is the key to making a problem easy to solve. In GSS, one selects the best breakout of resources (state subvectors) to simplify the processes (transformations of state).

We expanded that concept to the generalized state vector, one that consists of general information, not just variables that take on numeric values (be it integer, real, or complex). In particular, we consider that a GSS resource is equivalent to a data vector containing states such as RED, YELLOW, and GREEN. The data can be English words or character strings, as well as numbers. A generalized state vector may consist of one or more subvectors, i.e., GSS resources.

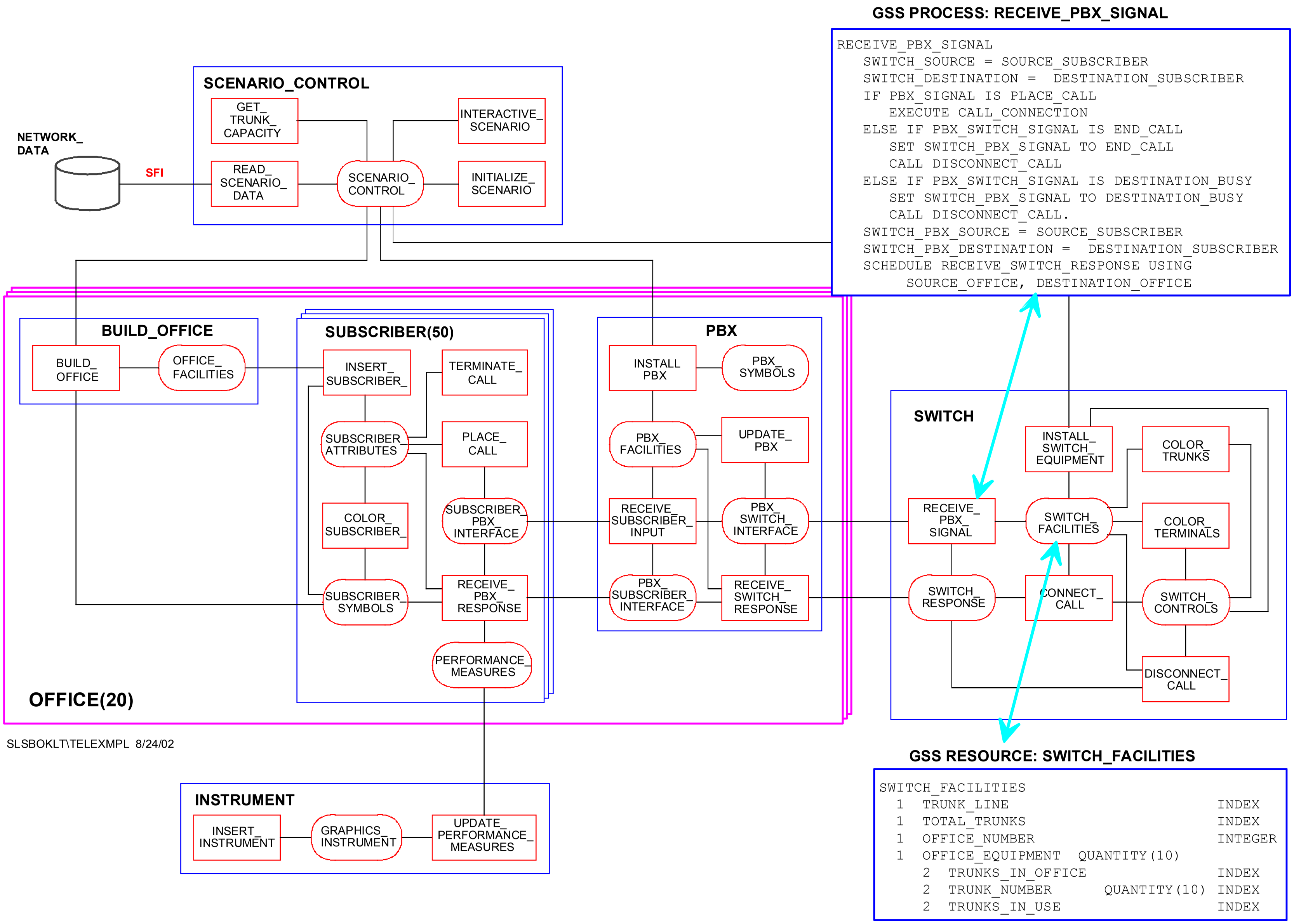

Figure 4-1 Telephone Network Models In GSS

Figure 4-1. Telephone network models in GSS.

With the above in mind, consider that a GSS simulation consists of a very complex *simulation state vector* (the simulation's data base) that changes as processes are invoked. At any instant of time, the simulation state vector contains the values of all the resources in the simulation. It must also contain the simulation queue, the simulation clock, and the real-time clock and random number generator seed if used. External files are considered inputs to an inhomogeneous model as described in [6], and are not part of the model's state vector.

State Space Definition Of A GSS Model

A GSS model only has access to a subset of the simulation state vector when a process in that model is running. We will call this the model state vector. A subset of the *model state vector* contains those resources that are actually part of the model. The state space representation of GSS is shown in Figure 4-2. The state vector that a model has access to consists of the following items and their corresponding information elements:

- ACCESSIBLE RESOURCES - The information contained in the resources that a model has attached to it. Note:- The SHARED resources may or may not reside within the model!
- SIMULATION QUEUE - The entries in the queue, including the indices it uses when it's processes are scheduled.
- SIMULATION CLOCK - The time of the simulation clock, including priority, if and when it schedules another process.
- REAL TIME CLOCK - The value of the real-time clock if and when it uses the real time clock.
- RANDOM NUMBER GENERATOR - The value of the current random number generator seed if and when it uses the random number generator.

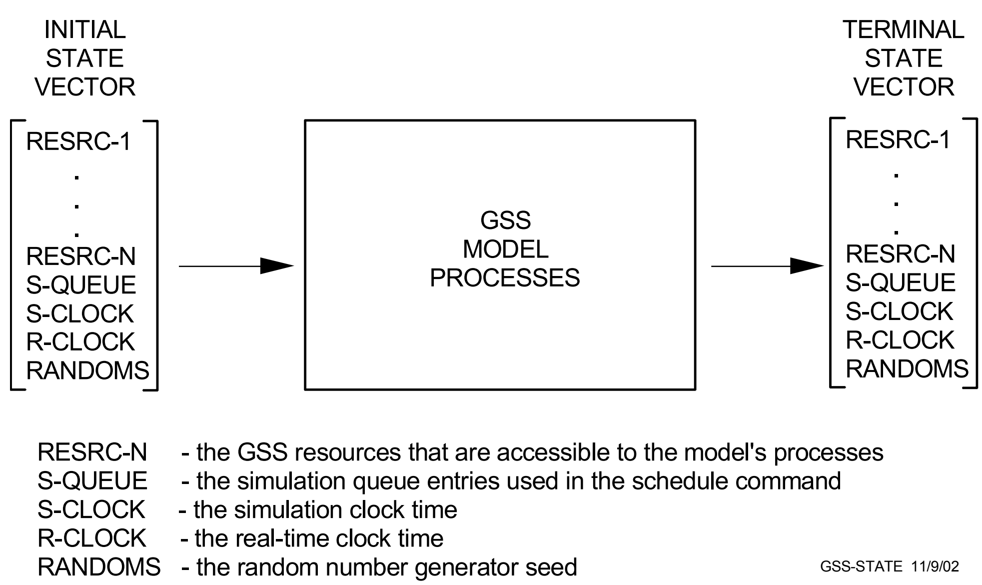

Figure 4-2. State space representation of GSS.

GSS processes are scheduled based upon the logic within itself or other processes. When a GSS process *runs*, it can schedule itself or other processes at specified times in the future, or at the current time. GSS processes run in zero *simulated* time. At any time, the state of a model depends solely upon its state vector. When a process in a model runs, its *terminal state*, i.e., the value of its substate vector - when it passes control back to GSS - depends solely upon its *initial state*, i.e., the initial value of its substate vector, and the rules within the process. When processes in another model share a part of the state vector of a given model, then any future state of the given model is, in general, dependent upon the rules in the other model, since they can change the given model's state vector.

Analogy To Symbolic Models Using State Space

The state space representation of a GSS model, Figure 4-2, is analogous to a set of differential equations that represent the state of a dynamic system at any instant in time. All future states are represented by the *equations of motion* in state space notation, and the initial conditions, reference Gelb, [23]. Electrical engineers have become accustomed to a graphical representation of the differential equations of electrical circuits, using interconnected icons of resistors, capacitors, inductors, generators, transistors, transmission lines, etc., refer to Figure 4-3. Such a drawing defines the differential equations of motion of the changes in electrical voltages and currents in the circuit. Given the initial conditions, the state of the circuit is defined for all time thereafter. In other words, the total dynamical description of the network is defined by the symbolic network.

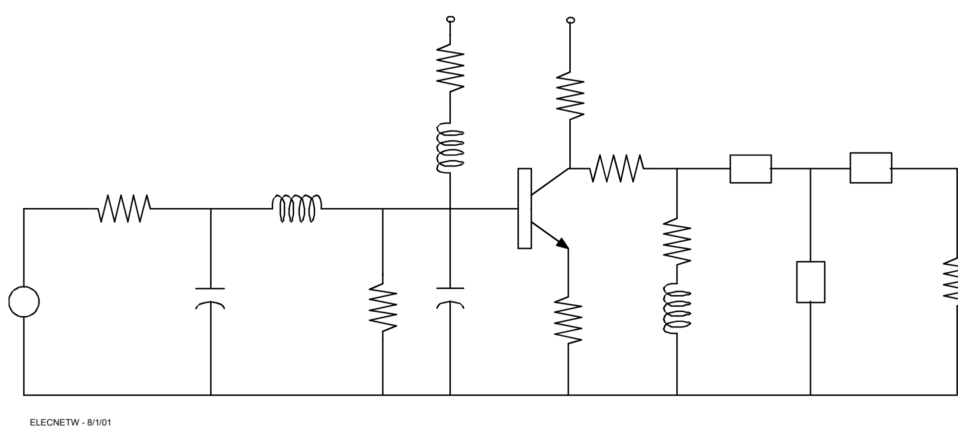

Figure 4-3. Iconic representation of an electrical network.

In GSS, the interconnection of resources and processes, as shown in Figure 4-1, is analogous to the electrical circuit drawing in Figure 4-3. Each has its corresponding *rules* and *storage* underlying each primitive element. In the case of electrical circuits, there are constituent equations that describe the changes in energy storage in differential form for each primitive icon. Representation of any system element must conform to this form of change.

In the case of GSS, sets of rules operate on sets of attributes (contained in data structures) to define the elementary change relationships in a model. Using GSS, the engineering drawing shown in Figure 4-1 and the underlying rule and data structures, define the total state of the simulation at any point in time after the initial conditions. This is known as the *generalized state space framework*.

Choosing the Most Convenient Reference Frame

As described above, the generalized state space framework, as implemented in GSS, supports the representation of discrete event systems as well as discrete time and continuous. Figure 4-4 illustrates that generalized state space provides the underlying framework for representing dynamic systems.

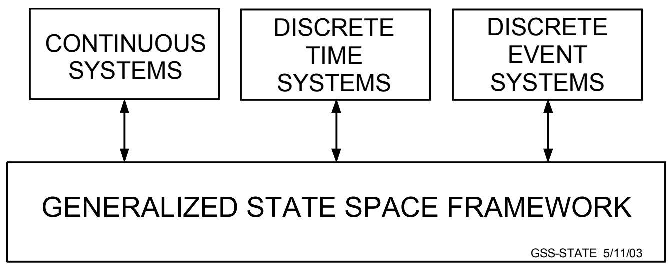

Figure 4-4. Generalized State Space:
- an underlying framework for representing dynamic systems.

The difference between representations of a system's dynamics is a matter of convenience (or maybe survival). A particular representation can be selected to support the economics of analyzing or predicting specific system behavior. If a system is conveniently represented by a set of differential or difference equations, then one of those representations might be best. If the system is more easily described by sets of rules operating on sets of attributes, then that representation should be chosen.

Since the advent of the digital computer, people have moved from analytical methods for integrating differential equations to numerical methods, especially when the systems represented are either nonlinear or nonstationary. Fast numerical algorithms for solving stiff nonlinear systems typically use complex heuristic approaches. What is interesting is that these approaches can be implemented implicitly using GSS rule and attribute structures. As computers provide significantly greater memory and speed advantages, the space for solving problems is growing, alleviating restriction to numerical methods for solution, and moving rapidly toward heuristic rule based approaches using complex data structures. These approaches are compared in *Simulation Of Complex Systems*, [10].

Having selected GSS as the overall framework, the analogy then becomes one of selecting the best set of information vectors (GSS Resources) to represent the system attributes. Depending upon how the resources are selected and structured, the rules (GSS Processes) may be much more simple to understand, build, and modify. *This is determined by the independence properties of the architecture, i.e. the interconnection of resources and processes - not the code!*

Reusability Analogy

In the case of electrical circuit modeling, a transistor model may require a significant effort to build and validate. Once completed, that model can be shared in many different simulations, as well as hundreds of instances used in a given simulation. Similarly, for models built using GSS. Development and validation may require significant effort, whereupon a given model can be shared in many different simulations by different organizations, as well as appear in hundreds of instances in a given simulation.

Complex models of electrical elements, such as transistors and transmission lines, may be made up of the primitive elements, and represented by higher order symbols. One can *push down* on these symbols and bring up the primitive representations that show all the detail underlying the model. More complex networks, such as groups of digital circuits in the form of gates and flip-flops can be represented using another level of hierarchy. In this manner, complexity is pushed down to the level that one wants to see it, and removed from view when it only serves to cloud the picture. This aids in both the understandability and reusability of a model.

Similarly, one can represent complex models in GSS using a hierarchy of models, wherein higher level icons are used to represent the highest level of a model, and one can push down as many times as needed to get to the primitive layer. In GSS, the primitive layer consists of resources and processes. This also aids in model understanding and reusability.

An Alternative Approach To Generalized State Space

In 1987, Ramadge and Wonham, [11], described the need to use English words as states in a control system. They introduced the notion of *alphabets* to deal with these non-numeric states. Their finite-state machine approach is somewhat different than that of the generalized state vector, particularly in the implementation of models describing complex systems. However, it appears that the underlying effect of these two concepts is essentially the same. Although there are no journal publications on this method, the generalized state space approach is documented in copyrighted GSS User's Manuals and PSI books on model development going back to 1982 and 1983. It is believed that these approaches were conceived independently.

THE RELATION BETWEEN INDEPENDENCE AND DECOMPOSITION

Most systems, such as sensors, are typically decomposed into elements that operate independently. These elements are then coupled to other elements via information exchanges. Figure 4-4 illustrates this situation. Sensors G1 and G2 are controlled by ground station G. Sensors B1 and B2 are controlled by ground station B. In this example, the sensors are independent of each other. G1 and G2 are tightly coupled to ground station G. B1 and B2 are tightly coupled to ground station B. Ground stations G and B are independent, but tightly coupled to central control J.

This organizational structure, having objects independent from a communications and control standpoint, has to do with productivity and survivability of organizations. This will be discussed more in the next section. Our point here is that, given such a structure, it best modeled accordingly. The flow and processing of information is modeled most accurately (and easily) by decomposing the overall model into submodels that represent the actual structure, and relative independence, of the sensor system.

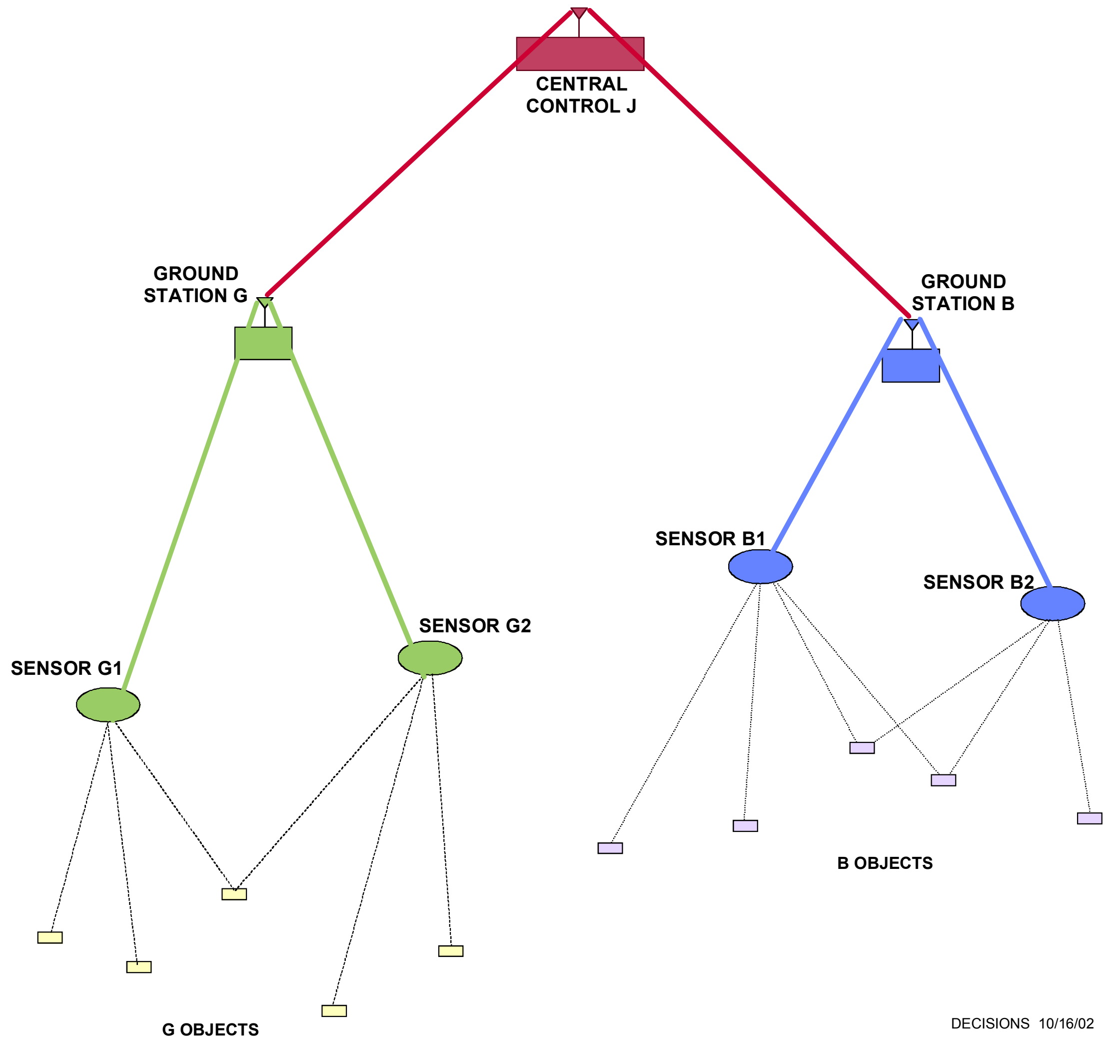

Figure 4-5. Illustration of sensor coupling.

In general, two objects may be organized to be:

- Independent - no coupling
- Loosely Coupled - only small behavioral changes will occur based upon information exchanged

- Tightly Coupled - large behavioral changes can occur based upon information exchanged

Organizational Decomposition And Control

Just tasking and tracking a single sensor can occupy a person full time. That person may derive information that is passed up the hierarchy to fuse with other data sources. Currently, many of the information exchanges are through human interaction. However, automation of both the systems and their interactions is a high priority for the Air Operations Center (AOC).

As problem complexity exceeds the ability of a single individual, it becomes necessary to divide up the work. This is particularly evident in a time-constrained situation. This is certainly the case in an AOC where hundreds of people occupy a single 8 hour shift, and three shifts operate 24X7.

Looking at Figure 4-5, let's consider some of the problems encountered in human organizations. If the organization at level 4 is representative, there are 155 people under the top (level 1) manager. If everyone worked in one spot, with everyone trying to talk to each other simultaneously, not much would get done. In the reverse case, if the top manager had to talk to everyone to make every decision, productivity would also be very low.

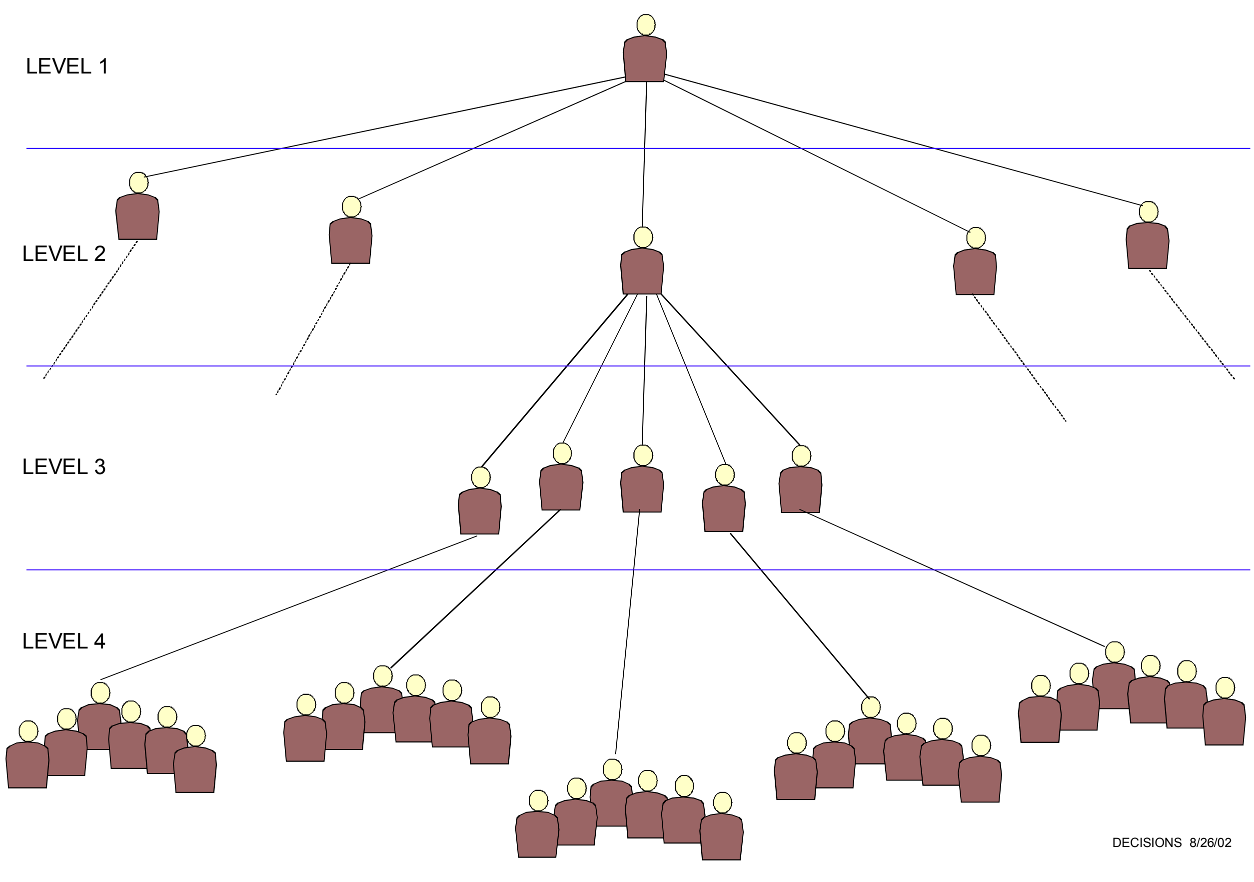

Figure 4-6. Illustration of organizational coupling.

There are people today that resist any form of control, and for whom organizational hierarchies are distasteful. These people attempt to sidestep the paradox that more than one person can have control at a time. This is the antithesis of leadership. These people are likely not operating in an organization in a competitive environment. When survival is at stake, these dreams disappear. Someone must have control. This does not imply that control cannot be distributed. This is done by distributing and delegating authority over specified elements in an organization. This is also done somewhat unconsciously as described by Arthur in [27]

Figure 4-5 illustrates this principle. Every leader has a span of control. If this span gets too large, productivity of the organization goes down. The leaders at levels 1 and 2 have a span of control of 5. Those at level 3 have a span of 6. This can be higher or lower depending upon the type of tasks to be done, and the capability of the people in the organization.

These organizational issues are not new to the military. When a decision must be made, someone must have both the responsibility and the authority to make it. Else productivity goes down. It is not by chance that military organizations have clear lines of responsibility and authority. It is because they are the organizations most concerned with survival. To help remove doubt, they wear symbols on their uniforms to signify their rank.

Organizations that follow rigid rules of independence may be less productive than others. There are good reasons to have coupling across organizations for information exchange. This coupling generally excludes control, and is therefore loose. Actions taken by an element of an organization may depend upon information exchanged across the structure as well as from a superior element. These links are normally well defined and approved up the hierarchy.

There are a number of points to be derived from this discussion regarding military organizations. These have evolved to maximize productivity in a typically scarce resource survival environment.

- Military organizations have evolved to be tightly coupled hierarchically, at least from a control standpoint, with relatively independent peer operations.
- Productivity increases as individual people or elements focus on specific tasks. This in no way precludes information exchanges, coordination, or teamwork.
- Cross coupling between peer organizations occurs, but is typically loose, resulting in actions based upon information exchanges as opposed to exercise of control.
- As the complexity of systems and equipment grows, the role of training in the use of specific systems and equipment has increased.
- This holds true for the opposing force as well as one’s own force. Therefore, people involved in intelligence gathering and fusion must be more highly trained in specific fields.
- The nature of planning, particularly in the AOC reflects the above bullets. Specific tools have been built to support specific assessment and planning functions.

Government, military, and commercial organizations are constantly hiring consultants and performing self-analysis to find ways to be more productive. Technology advances that afford improvements in productivity, e.g., computers, communications, and special equipment, are absorbed most quickly in highly competitive organizations concerned with survival. The U.S. military is a successful example of this.

Learning And Evolution

Our focus here is on improvement of the situation assessment and planning functions in the military, and particularly in the AOC. It is hard to envision a small group of people replacing the staff of a full scale AOC in the foreseeable future. It is possible to envision a much more productive organization, one that can simulate courses of action (COAs) and corresponding Enemy COAs (ECOAs). This will involve the ability to predict friendly and opposing force outcomes with reasonable accuracy over significant time periods with many unfolding events.

Accuracy of prediction will be a major factor in reducing losses of people and equipment while achieving the desired effects. One path to achieve this goal is to take advantage of the learning and evolutionary progress already achieved. There appears to be a lot to learn from the decomposition of military force structures and the corresponding independence of functions.

SIZE OF THE PROBLEM

The concept of boundedness was described in Prediction Theory, [6]. It implies that we were concerned with finite time periods, finite sets of data, and bounded values. In this context, finite and bounded meant *not infinite*. It meant that certain mathematical properties existed as long as numbers did not go to infinity. It had nothing to do with *small*. In addition, it provides for state variables represented by real numbers, i.e., they can take on an *infinite* number of states.

The concept of boundedness must be contrasted with techniques, e.g., finite state machines, that typically depend upon small numbers of state variables for practical implementations. Small in this context might imply numbers less than 10^6 (if all the variables were real, this only takes 4 megabytes of memory). GSS has been used to deal with numbers much greater than this in detailed models of communications. When one provides for platforms, sensors, C2, weapon systems, etc., the number of state variables could rise to 10^{10} (taking 40 gigabytes of memory). We note that the set of values of state variables can be virtually infinite since they are real.

Having modeled complex communication systems, one learns that the devil is in the details. The butterfly effect is alive and well. Military communications are fraught with highly correlated traffic. One or two target sightings can set off a huge set of events. The GSS schedule queue PSI can support 10^6 events in at any instant of time. GSS has already been used to model 1000's of complex radios (e.g., SINCGARS, EPLRS, JTIDS, or JTRS) in a simulation, each with 10 to 100 events in the schedule queue at an instant of time. These simulations have been very accurate at predicting network performance using highly correlated traffic.

One can argue that understanding what is going on in such a complex simulation is difficult. However, with a suitable architecture and sufficient graphics, the number of subject area experts required to understand what is going on can be accommodated. There can be many people tied in, concurrently watching results and preparing inputs. Accordingly, we envision a decomposition of the functional architecture much along the lines of the existing functions of the AOC - yet fully integrated.

DRAWING THE LINE BETWEEN HUMAN JUDGEMENT AND AUTOMATION

How do we decide what's best for a human to do versus using a computer to do it? This problem has been addressed for many years in the field of CAD. One must answer the questions: What processes depend upon - or are best left to - human judgment? Where are the break points where computers do better?

It generally comes down to time. Given the requirement to achieve a given quality of results and objectives, e.g., being able to meet specified accuracy requirements, how much time will it take to get from here to there. To restate what was said above (in Chapter 4), when building tools to help people solve design problems or make complex planning decisions, time enters into the picture in at least two major ways.

- Development Time - the time it takes to develop a tool
- Solution Time - the time it takes to get useful solutions from a tool

There is no fixed answer. Both of these times depend upon the state of technology. However, there are general principals that apply when trying to decide upon an approach. These principals assume agreement upon the development time and solution time requirements.

Figure 4-6 provides an illustration of how the level of automation achieved tends to grow in various applications. Some applications have achieved a high degree of automation quickly. These tend to have a high degree of rote functions. Some have to wait for technology to catch up the be practical. Others have clear limits in terms of % automation, at least with foreseeable technology.

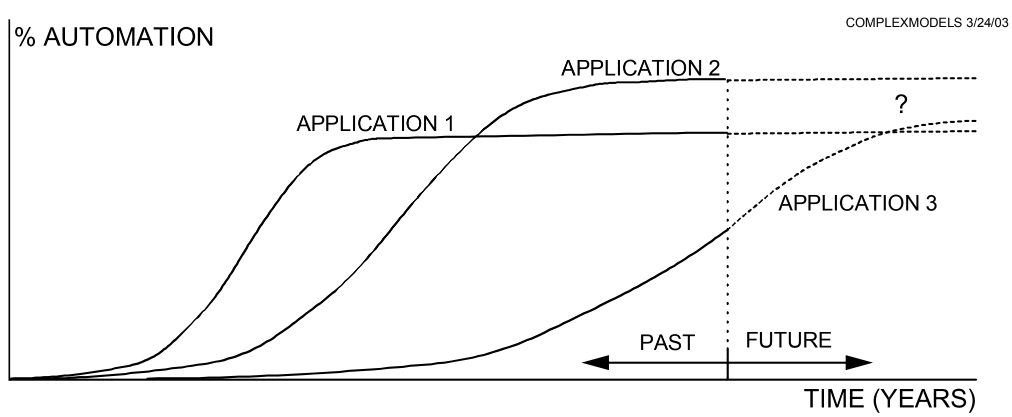

Figure4-7. Level of automation achieved (past) and predicted (future) for various applications.

5. STOCHASTIC NATURE OF THE PROBLEM

There are various levels of planning in a real battlespace. For example, one may have to determine the "best" approach to moving troops or supplies from one location to another. This can be treated as a classic transportation problem. Given the knowledge of available transportation facilities, air, sea, and ground routes, and the myriad of other factors affecting the time and energy required to complete the move, one can apply standard techniques, e.g., Linear Programming (LP), to come up with the *best* solution. But is *best* good enough?

Dealing With Variations

The LP approach is excellent for coming up with answers to the problem as posed. However, by itself, it does not deal with the stochastic nature of such a problem. In reality, every attribute can be subject to variations. In the transportation problem, these variations can be due to traffic, weather, breakdowns, etc. These variations can be taken into account in various ways. For example, traffic may be predictable based upon day-of-week and time-of-day. Even so, traffic can get tied up due to special events. The time to get from point A to point B can be adjusted by a traffic variable. The traffic variable can be a function of the calendar and clock. Actual traffic can also vary around a mean value due to effects that appear random, and are therefore unpredictable. These variations must be accounted for when determining whether the solution meets the time constraints.

To generalize the approach to characterizing traffic, each route may have variations in time that can be broken into two categories, those that are predictable based upon observable attributes, and those that appear to be random. If we can develop relationships between the predictable variations and the observable attributes, they can be applied to adjust the mean value. This serves as additional information to reduce the prediction error.

There are approaches for characterizing the effects due to the random variations. The approach used most often is Monte Carlo Analysis. In this case, distributions are postulated for all of the random variations. Then a simulation is run with random samples drawn from these distributions each time an event occurs requiring a value for the variation. Depending on the scenario, one runs enough simulations to characterize the distributions of the resulting measures of performance. For example, total time to move the troops from A to B may involve many traversals of the many routes. When these individual traversals are simulated, they are subject to the variations determined by the random samples. If a new random number seed is used for each simulation, different results will occur for the total time measure.

After enough simulations are run, a histogram or other measures of these times can be used to characterize the distribution of total time. Consider Figure 5-1 as the resulting distribution representing the time to move troops from A to B. If the simulations took into account all of the variations present in the real battlespace, then one can derive a probability statement about the range of time. For example, if Tmax is 20 hours, and the area under distribution D1 up to Tmax is 95% of the total, one can state that troops can be moved from A to B in 20 hours with a 95% probability.

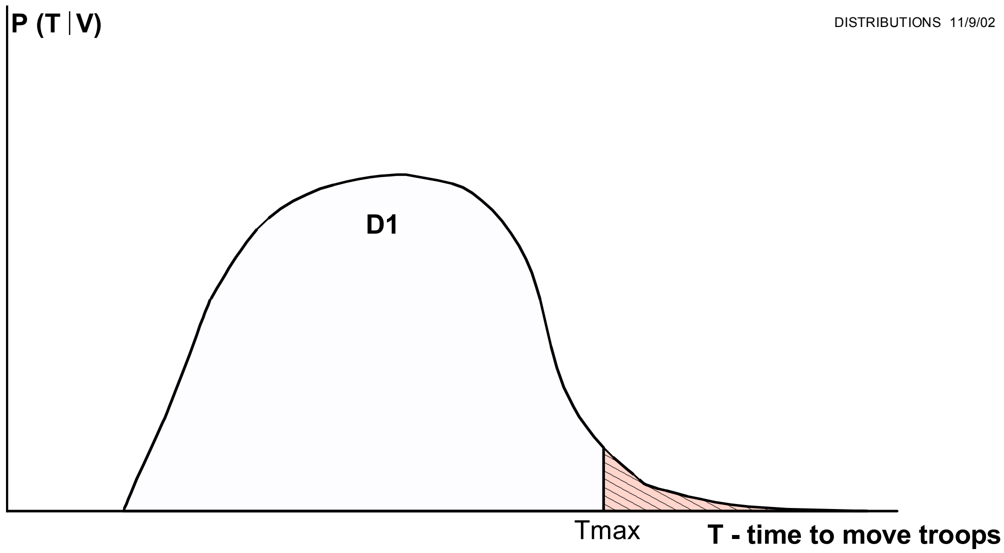

Figure 5-1. Example of a measure of performance characterized by a distribution.

The probability statement comes directly from a mathematical calculation based upon the distribution. If the distribution represents the real world perfectly, then the probability statement is correct. One must ask how accurately the distribution represents the real world. This is answered by providing a confidence level in the distribution relative to the calculations being used. This is also described in [6]. In addition, there is a more direct way to get to a solution without using Monte Carlo. This is described below in Accounting For Constraints.

Dealing With Large Decision Trees In Time

Military planning requires a large number of decisions to be made over time. In many cases, large numbers of decisions must be made to start or continue an operation before any results can be seen. Most of these decisions involve selection of an approach from many choices. Each of these decisions lead to the next decision level where more selections must be made. Just considering the sequence of decisions coming from each level in the command and control hierarchy, one could envision a very complex picture of this process similar to that in Figure 4-5.

As results start to come in, decision makers, with the help of their staffs, must start to characterize the adversary's changes in capabilities, plans and decision processes. At each step along the way, at different levels in the decision process, the characterization of effects achieved can be represented by a distribution as in Figure 5-2.

Although Figure 5-2 looks like a *normal* distribution, a large number of variations that one may be faced with are not characterized. To use a Monte Carlo approach, or the worst case design approach defined below, the distributions can be unknown but bounded. If the Tmax boundary in Figure 5-1 is known, i.e., it is the 95% point, that's all we need to know. We need not know the shape of the distribution. But in many cases we don't even know that.

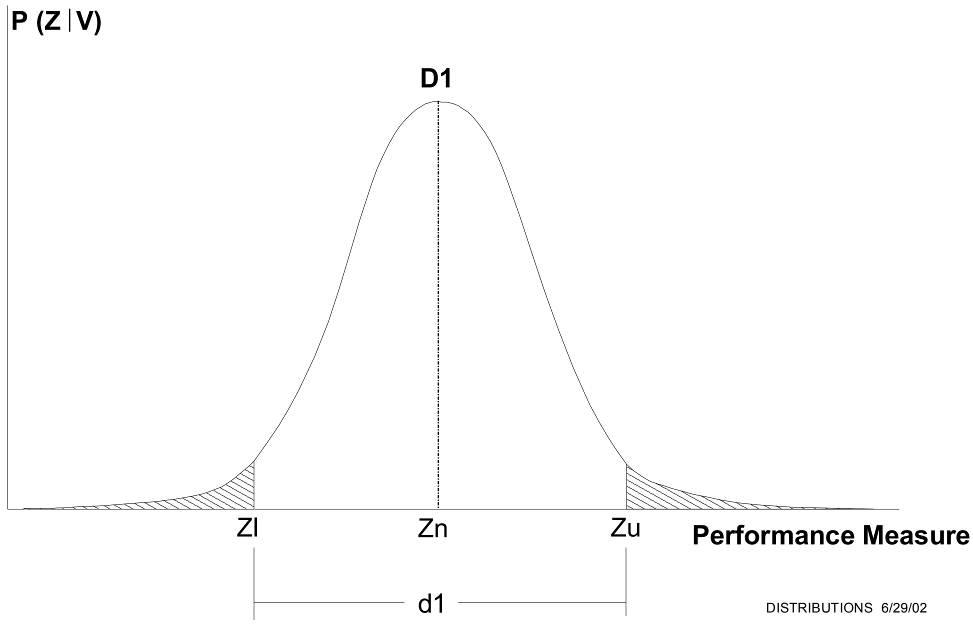

Figure 5-2. Desired effects characterized by a distribution.

Considering all of the possible variables and characterizations, one may feel overwhelmed by what would appear to be unpredictable chaos. But operations do unfold according to rules. The rules may be changing, but some level of rules and coordination is required to achieve a desired level of effectiveness.

Accounting For Constraints

The need for rules and coordination in operations imposes constraints on behavior. This need increases with the tempo of operations. In addition, real world systems are nonlinear, imposing additional boundaries of constraint. Behaviors never get to infinity. Something breaks down first. In addition, there are different levels of conflict and corresponding missions, and these can be bounded in terms of their outcomes in time. Figure 5-3 illustrates the envelope generated by successive distributions in time.

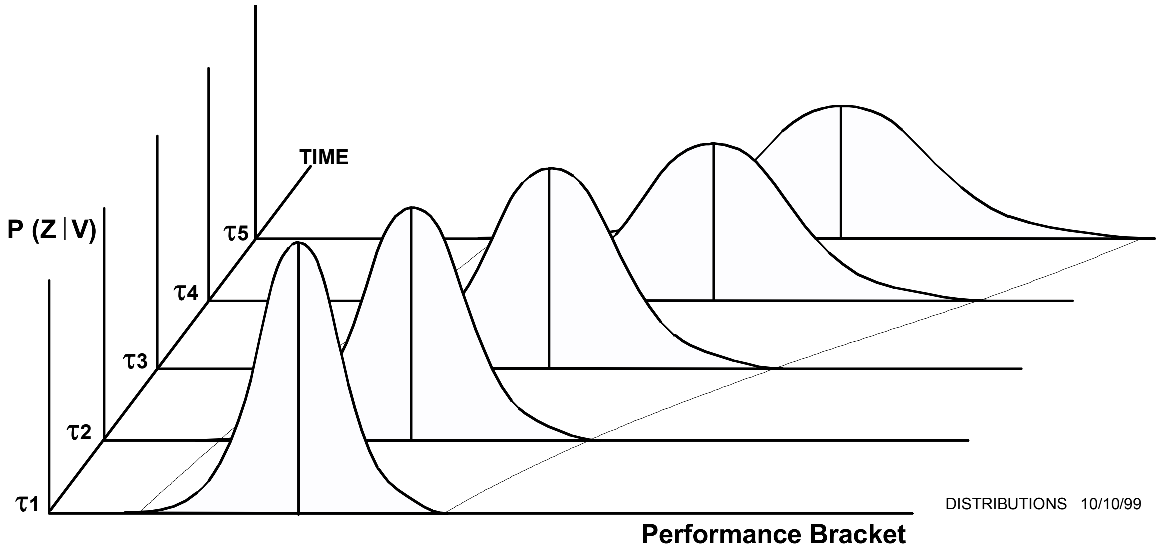

Figure 5-3. Prediction of effects characterized by a distribution envelope.

Analyzing the potential manner in which events unfold in time, and the way they contribute to variations in potential outcomes, provides an improved understanding of how one may want to proceed to reduce the risk of unnecessary losses. Commanders can lay down operational constraints, e.g., *this mission will not expend more than some specified amount of fuel or ammo*; or, *this flight will concentrate on that target*. All of these constraints serve to bound the problem.

6. FINDING FEASIBLE SOLUTIONS

Finding feasible solutions implies finding feasible COAs or sub-COAs. These are actually trajectories in time. We will start by looking at a single time point.

OBJECTIVES AND CONSTRAINTS

Problems of this nature can be defined in terms of objectives and constraints as described in the linear or nonlinear programming literature. In practice, the constraints are usually more important than the objective function in that they must be satisfied to provide a feasible solution. This translates to a COA that satisfies prescribed constraints to be acceptable. Examples of such constraints are: risk of casualties or loss of life; limits on platform availability; limits on personnel; limits on ordnance, limits on fuel, limits on time, flight path restrictions, jamming restrictions, frequency allocation restrictions, etc. These are referred to as hard constraints, in that a violation of any such constraint renders the COA unacceptable (the solution infeasible).

The constraints of interest here can be mapped into the parameter space of vector variables, V, that determine the constraint surfaces. Constraints can be posed in terms of the variables in this space such that a constraint function H is positive when the constraint is satisfied and negative when it is violated. Figure 6-1 below illustrates such a mapping for four constraints.

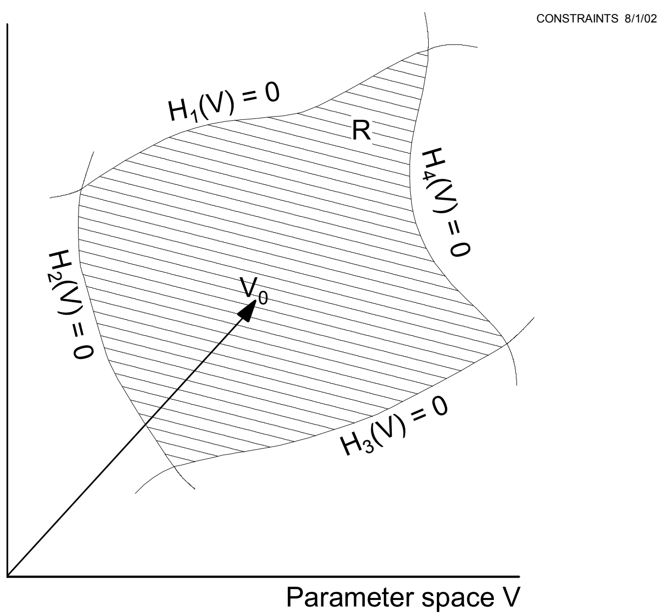

Figure 6-1. Hard constraints defined by surfaces Hn(V) = 0.

By visualizing the surfaces Hn(V) = 0 (the curves bounding the region R in Figure 3), one can interpret this geometrically. A solution, Vo, is defined as *feasible* if it satisfies all the constraints. If all points inside the region R ensure that all of the values of the Hn are greater than zero, then R is defined as the feasible region, bounded by the constraint surfaces Hn(V) = 0.

We note that this illustration is for an instant of time. In fact, Figure 6-1 is just a snapshot from a trajectory in time, showing only the spatial parameters. We will limit our discussion here and deal with trajectories in time in a later section.

As the solution vector, Vo, moves outside the region R, at least one of the constraints is violated. The feasible region appears large in Figure 3. However, this is done to illustrate the definitions. In the practical problems of interest here, the solution vector will depend upon many parameters that affect at least one of the constraints. For example, a constraint on the probability of being engaged by an IADS missile will depend upon where one travels in x, y, z space. A constraint on fuel will depend upon altitude and distance traveled, a function of the way points of a flight path in x, y, z. It will also depend upon the position of refueling tankers - a different set of parameters. The dimension of the space can be quite large.

In addition, the constraint surfaces can be very nonlinear functions of the parameters. In a large parameter space, it may take time to develop the complete set. There have been studies of such constraint surfaces for similar problems, and it is known that they can take on exotic shapes. This can make the feasible region very small. Depending upon how the problem is posed, it is not unusual for it to be non-existent.

ACCOUNTING FOR PARAMETER VARIATIONS

To further complicate the problem, the actual values of parameters will vary. For example, a tanker may have been tasked to remain in a certain small area, but circumstances forced it to go to another area. Targets may move before they're engaged. IADS radar coverage may not be known precisely. All of these will cause a flight to change its way points. This implies that we only know the value of the solution parameters in time to within a distribution. We may not know the shape of the distribution, and may only have some knowledge of its bounds - in terms of percentile limit values. This is the classic worst-case design problem. We will not delve into the details here, but will outline it and provide references for detail.

Figure 6-2 illustrates what happens when these parameter variations are taken into account. If Vo is the selected (nominal) solution, and we apply all of the possible variations, T, out to a selected set of limits on their distributions of each parameter, a region $\mathbf{r_0}$ will be described as shown. Thus, $\mathbf{r_0}$ is the region of all possible values of the actual solution, V. This implies that all of the points in $\mathbf{r_0}$ must remain in R, else a constraint will be violated.

This problem has been solved using Computer-Aided Design (CAD) techniques, see [14], [15], [16], [17], and [18]. If for each point on a selected constraint boundary we allow the T vector to take on all of its possible values, we will describe a manifold about the original constraint boundary. This is shown in Figure 6-3 with two curves on either side of the original boundary of R.

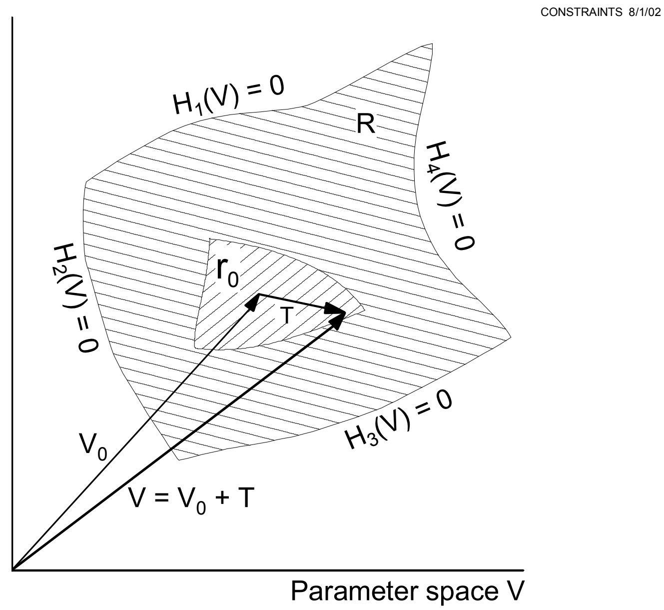

Figure 6-2. Possible variations of the solution due to parameter variations.

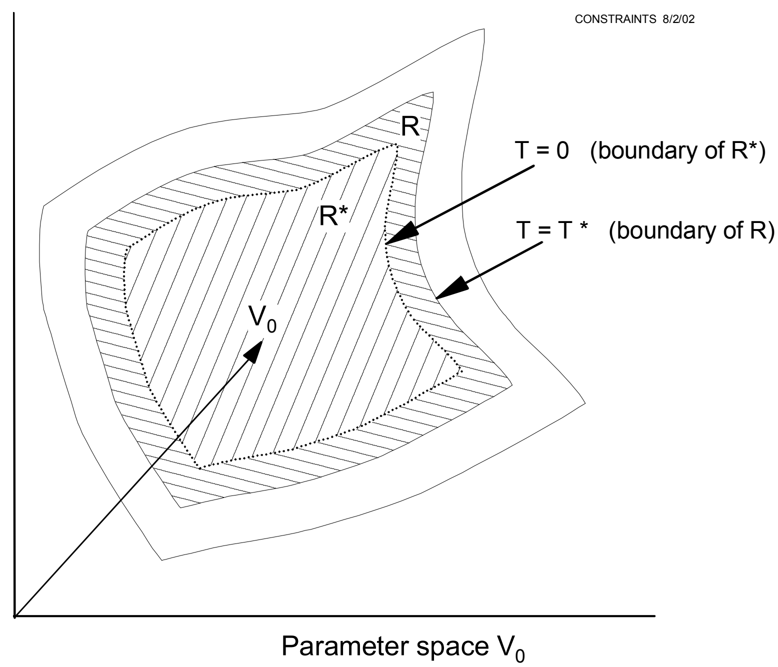

Figure 6-3. Transforming the constraint boundaries using optimization.

Alternatively, for each point, Vo, on a selected constraint boundary we can determine the value of T = T* that causes H(Vo + T*) to be most negative. These values form the inner region R* bounded by the surface H(Vo + T*) where T* depends upon Vo. This approach conveniently transforms the original feasible region into a smaller region such that, if the solution Vo falls within the transformed region, R*, it will meet the worst case conditions. This is described in further detail in [14].

WORST CASE DESIGN OPTIMIZATION

A major benefit of this approach is that it supports direct optimization and therefore synthesis of a solution that meets worst case constraints. One avoids the iterative approach of finding feasible solutions and then running Monte Carlo analysis to determine if constraints are violated. It is a proven technique that has been used extensively in very difficult design problems.

Before considering optimal solutions, one must further investigate the worst case problem, i.e., searching for the feasible region after applying the worst case transformation. In practice, it is not unusual for the transformed feasible region to be null, i.e., the feasible region has disappeared. This implies that the problem as posed cannot be solved without violating one or more constraints.

In this case, one must go back and rethink the problem, and this usually means relaxing one or more constraints to get a solution. One can then look at this solution and determine what constraint is hard to meet. Alternatively, with good optimization techniques, this information can be produced as a by-product of the feasible search process.

Before moving on to the optimal control problem in the time domain, there are two other difficulties that must be considered when attempting to solve highly nonlinear constrained optimization problems. First, it is not unusual to find multiple disconnected feasible regions. This means that the optimization algorithms must be able to seek out these regions for better solutions.

Second, solutions may be unstable. In the case of a feasible solution, this occurs when the solution is very close to a constraint boundary. This implies that a very small change will render the solution infeasible, i.e., one or more constraints is violated. At this point, judgment must come into play. Either these conditions must be anticipated and accounted for in advance, or a decision must be made on the spot. Again, with good optimization techniques, this information can be a by-product of the feasible search process.

Another form of instability occurs with optimal solutions when the objective function has very narrow peaks. Again, with very small changes in the solution, large changes can occur in the value the function being optimized. In a very nonlinear problem, this can be significant. All of these difficulties can be accounted for and alerted using good optimization techniques. These will be illustrated in the next sections.

As indicated above, a COA is a control sequence implying a trajectory in time. In the constrained optimal control problem, if there is a feasible region, the control sequence (and resulting trajectory) is bounded by a constraint manifold in time and space. Instead of finding a solution to stationary problems as described above, one must find a sequence of steps that weaves a trajectory through this manifold without going out of bounds. This is illustrated in Figure 5-3. Clearly, this is a much more difficult problem to pose and solve. In fact, this is the problem - finding a COA that weaves through the constraint boundaries without a violation.

7. MISSION PREDICTION EXAMPLE

We will evolve our proposed solution approach by way of examples. We will start with a mission level example to keep the complexity at a level we can deal with. Clearly, the mission level example we will use, albeit complex, leaves out important considerations at the campaign level. Probably the most significant is the human behavior factor. However, there are many important factors accounted for in the mission planning problem that we will address first.

Figure 7-1 depicts the major aspects of the mission planning example. Basically, a mission is planned to fly from the blue dot to targets near an air field, and back to the blue dot. The problem is to determine the best set of way points that maximizes the probability of neutralizing the targets, while meeting the constraints of getting pilots and aircraft back without loss.

Measuring Mission Success

We start with definitions leading to measures of mission success. The end-state of success can be represented by discrete or continuous outcomes that depend upon one or more state variables. Defining Φ as the measure of success, e.g., the probability that targets are neutralized.

$$\Phi = \Phi(x_1, x_2, ..., x_N) = \Pi(\text{Targets Neutralized}) .$$

Figure 7-2 illustrates different ways to measure the probability of success, Φ. Figure a shows the probability of taking on two discrete states signified by YES and NO. Figure b shows a continuous probability on a continuous set of outcomes in the range [0.0, 20.0]. We note that in case b, both the range of the Random Variable (RV) and the probability are infinite sets.

In the mission planning scenario, there are two targets A and B. One can measure the resulting damage in terms of the level of incapacitation of the targets. One can make Φ a function of two state variables, each providing a numeric measure of the incapacitation of each target. Various measures can be used to increase the value of hitting a particular target, or hitting both targets.

The mission will be planned initially with 4 aircraft, two with appropriate ordnance, and two with EW support systems. The problem is to maximize Φ, target incapacitation, while meeting the constraints that the aircraft return to the RF point safely. We will now discuss the constraints.

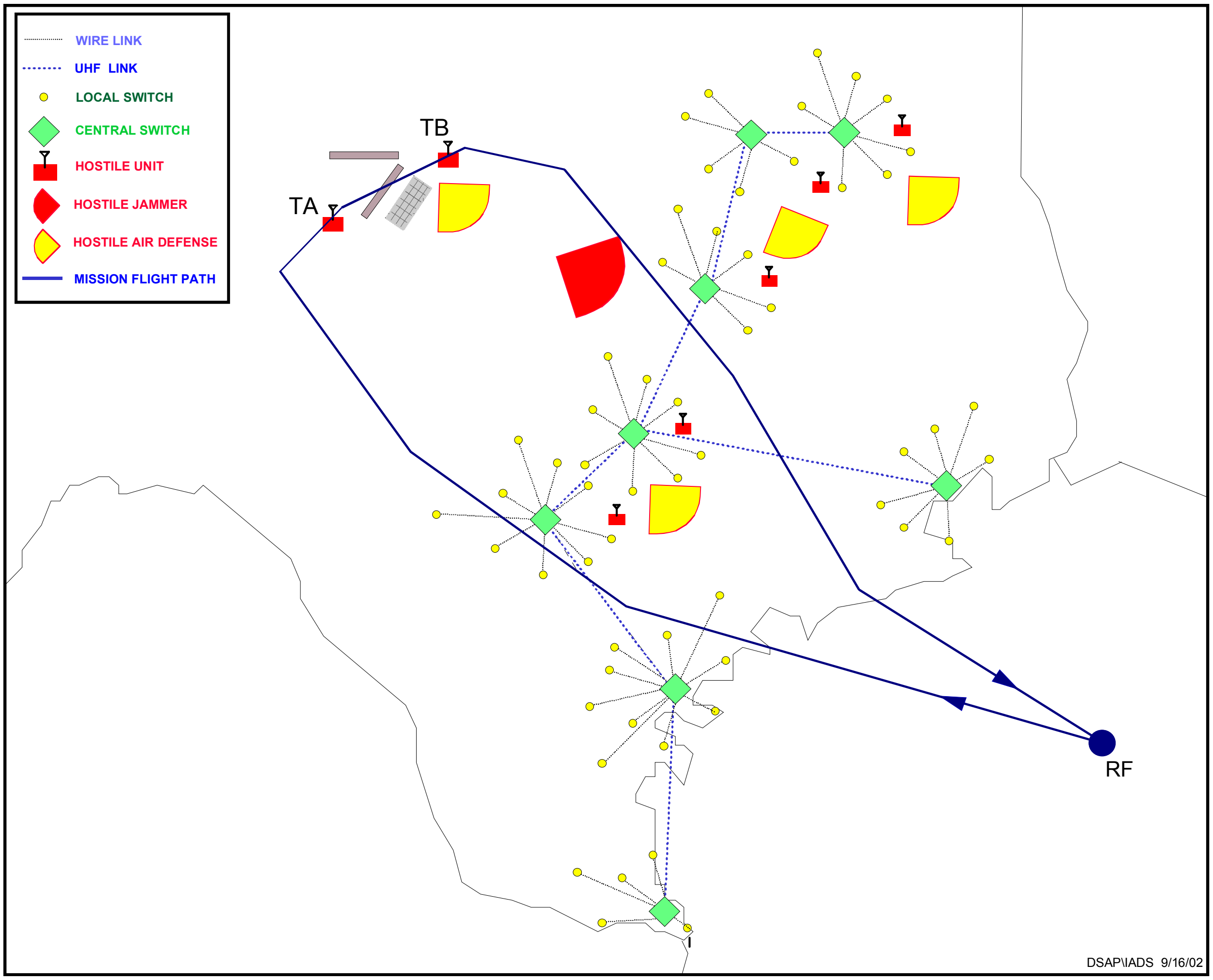

Figure 7-1. Mission optimization example.

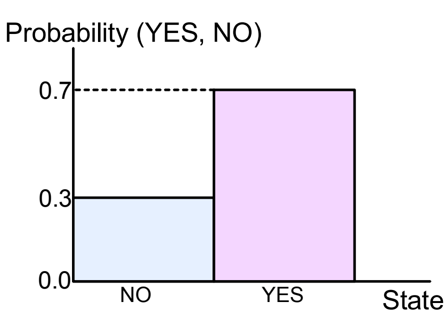

Figure 7-2a. Discrete states.

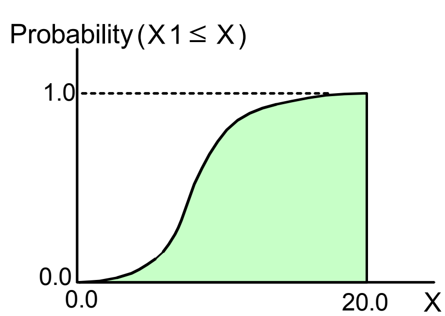

Figure 7-2b. Continuous Random Variable.

We will impose the constraints that each aircraft gets back safely to point RF. This imposes four constraints, one on each aircraft.

$$H(1) = H1(x_1, x_2, ..., x_N) \geq 0 \quad ==> \quad \text{Aircraft 1 returns OK}$$

.

.

.

$$H(4) = H4(x_1, x_2, ..., x_N) \geq 0 \quad ==> \quad \text{Aircraft 4 returns OK}$$

Determination Of The Constraints

The constraint functions will be posed in terms of probabilities. This is similar to a reliability problem. We will set the constraint such that the probability of losing an aircraft is less than 1%.

$$\Pi(\text{lost}) \leq 0.01$$

This constraint must be satisfied before the mission plan can be accepted. Rewriting this constraint in terms of an H function that must be positive,

$$H(1) = 0.01 - \Pi(\text{Aircraft 1 lost}) .$$

This implies that the constraint will not be met (is positive) unless the probability of losing Aircraft 1 is less than 1%. We must now determine the probability of losing Aircraft 1. We will interpret this as the aircraft being shot down or running out of fuel.

$$\Pi(\text{lost}) = \Pi(\text{shot down} \cup \text{fuel out})$$

The evaluation of this probability, as well as the measure of success of the mission, will be accomplished by models in the simulation. Some of these models will be used to predict red COAs. Based upon the requirements defined under Worst Case Design Optimization in Section 6, it may be necessary to evaluate the *realistic* worst case red COA. Coming up with realistic worst cases can serve to simplify the problem. This is discussed in examples below.

Selecting The Best Flight Path

We must now select a flight path that meets all the constraints and provides the highest measure of success. This is done in GSS using the nonlinear constrained optimization system. Using this system, a set of ranges on the coordinates of each way point is imposed by the analyst. The positions of the targets, the air defense radars, the SAM sites, and the antennas for the UHF communication links are known within some specified error distribution. The system poses a set of way points, and the platforms fly a mission using these way points. As missions fly, ground radars may pick them up and send track data to C2 centers. C2 centers can assign weapon systems to engage the aircraft. The weapon systems will determine which targets to engage based upon the rules of engagement for that weapon.

The platforms contain models of early warning receivers, self screening jammers, and other EW gear on the aircraft. As they fly through the air defense system, a probability of being engaged is computed based upon the signals received at each radar, the communication between the radars, the C2 centers, and the weapon systems. When this probability exceeds the constraint threshold, the constraint is violated and the path is considered to be a nonfeasible solution. In addition, the fuel is checked. If the fuel is exceeded, then the constraint is also violated.

After a sufficient number of paths are tried, and a feasible solution is found, i.e., no planes are lost, then the system starts to search for feasible paths that increase the probability of incapacity of the targets. It is possible that the problem cannot be solved as posed. This generally implies a bad selection of way point ranges, or a different combination of aircraft. It is up to the analyst to formulate another approach to the mission.

TOOLS TO SUPPORT MISSION PLANNING

It is apparent that the mission planning tool described above requires inputs from other tools as shown in Figure 7-3. In fact there are a multitude of tools that aid directly or indirectly. Examples are the Tactical Air Mission Planning System (TAMPS), Portable Flight Planning System (PFPS), Joint Munitions Effectiveness Manual (JMEMs) and Joint Targeting Toolbox (JTT). In Figure 7-3, the mission planning process has been broken out into several elements, some of which require inputs from separate subsystems, e.g., from the ISR and IADS subsystems. It should be emphasized that many functions are currently performed at the Wing and Squadron levels. Other tools are DIODE, AFMAS, Falcon View, ..., etc.

The ISR inputs, as shown in Figures 7-4, are needed to support the IADS simulation as well as the mission planning process. For example, positions of targets, air defense radars, SAM sites, and antennas for UHF communication links must be known within some specified error distribution. If not, the IADS simulation cannot produce a valid probability statement.

Figure 7-4 also illustrates the distributed nature of ISR assets. Inputs from ISR assets can come directly from a sensor system, or from a higher level management system that has fused data. The connectors with INT inside are shown to indicate interfaces to other systems or tools, e.g., the IADS, the EP Simulation, etc.

IADS Simulation

The IADS simulation, Figure 7-5, is used to fly simulated missions to determine the effectiveness of a planned mission. It can be used to determine the outcomes of many trials. These missions can cause changes to the IADS as they are flown, in turn causing the resulting flight to be affected by the changes to the IADS. The models cover both red and blue assets.

The IADS simulation must contain more than a basic representation of the Integrated Air Defense System. It must contain all of the elements that affect the effectiveness of that system. This includes the logistical support facilities, e.g., electric power, POL, and delivery of personnel, ordnance, other goods, and services needed to maintain the IADS at a given operational level. The IADS simulation must take inputs that describe the state of these support elements to determine its ability to react to blue missions.

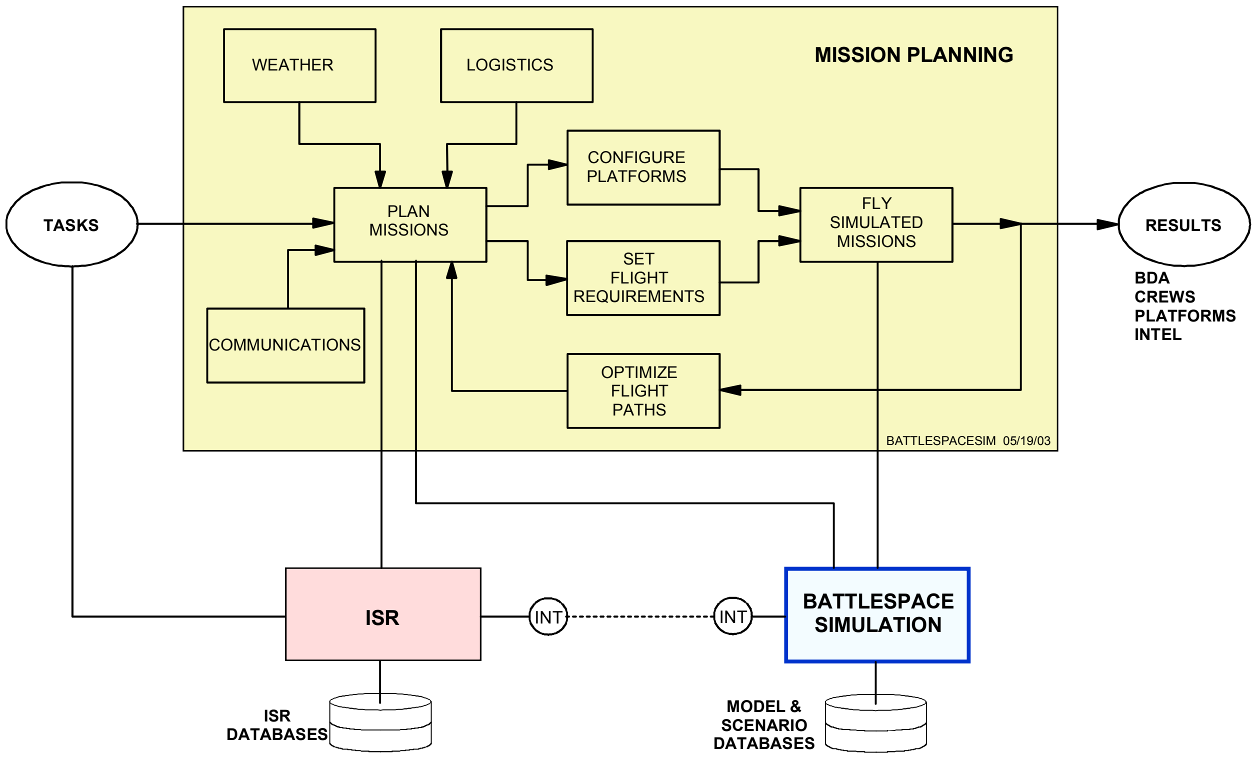

Figure 7-3. Inputs from other tools.

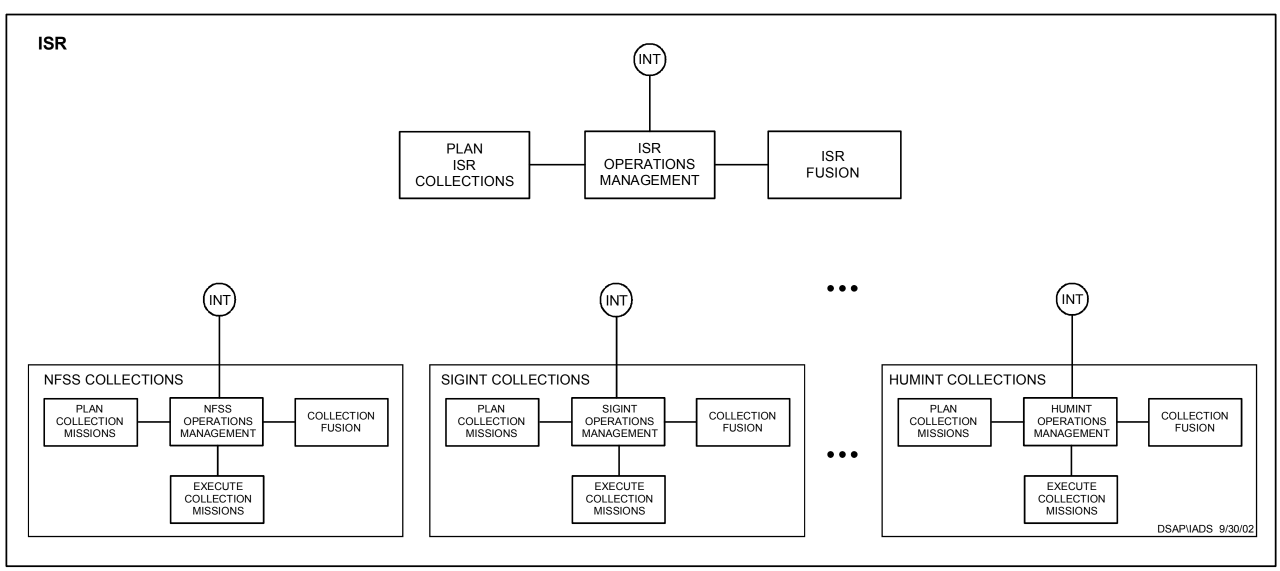

Figure 7-4. ISR inputs are required to support the IADS simulation.

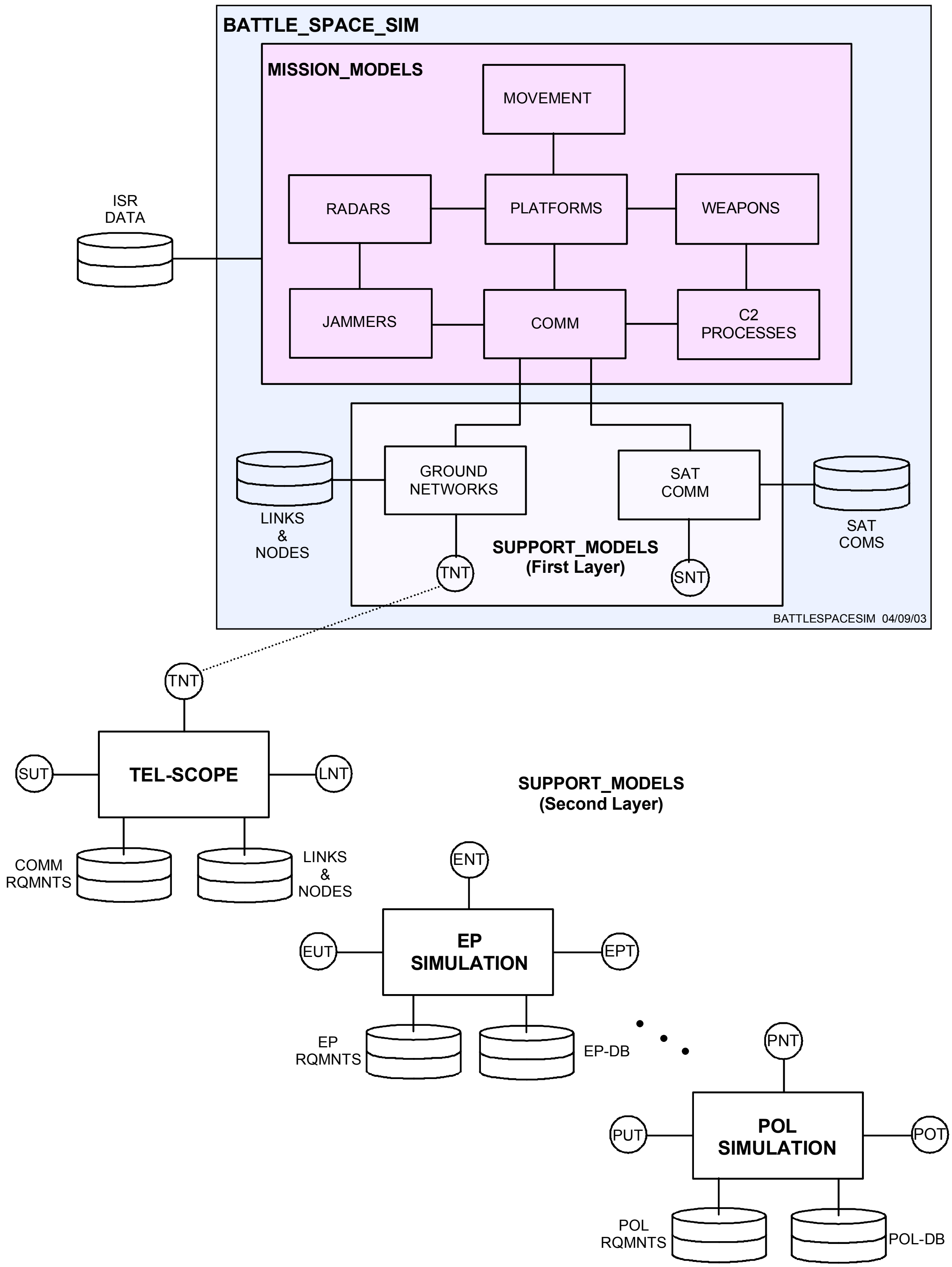

Figure 7-5. IADS simulation used to select missions.

ISR And Other Inputs

As indicated above, ISR inputs are essential to the IADS simulation and can be absorbed in real time. These inputs determine where entities are located, their operational capabilities, the level of support they are getting, and their rules of engagement. This is the key information that conditions the probability statements coming out of the IADS.

ISR inputs must be provided in terms of probabilities and confidence statements that can be used to project the probability of mission success and constraint violation. Position locations are typically given in terms of the Elliptical Error Probability (EEP) or Circular Error Probability (CEP) around an estimated location. These inputs are critical to increasing the accuracy of predicting the various probabilities of possible outcomes during a mission.

Note, the ISR subsystem can be treated as distributed, instead of going through what may appear to be a bottleneck. For example, many systems want direct output from the Army's Netted Full Spectrum Sensor (NFSS) Operations Management System (OMS). They will not want to go through a higher level manager. The NFSS will also be getting tasking requests and other inputs from these systems directly. A sensor management system at a higher level may have some degree of control, but this will likely have to evolve and be reconciled on a Joint Services basis. Therefore, the IADS will likely have to get its inputs from multiple sensor management systems directly, at least for some time to come.

MISSION TRIALS

Looking at Figure 7-6, the mission planner (currently at the Wing or Squadron level) may start by configuring a single aircraft with minimal ordnance to take out the required targets. He then runs the IADS simulation using optimization to find an optimal flight path to get these targets. Depending upon the mission and the results, he may have to go through several trials as indicated below before achieving a successful result.

1. Configure aircraft with minimal ordnance to get targets. Run IADS and evaluate results.
2. If mission is not successful, reconfigure aircraft with self-screening jammer. Run IADS and evaluate results.
3. If mission is still not successful, reconfigure aircraft with additional protective ordnance. Run IADS and evaluate results.

   .

   .

   .

9. If mission is still not successful, configure additional aircraft with ordnance. Run IADS and evaluate results.

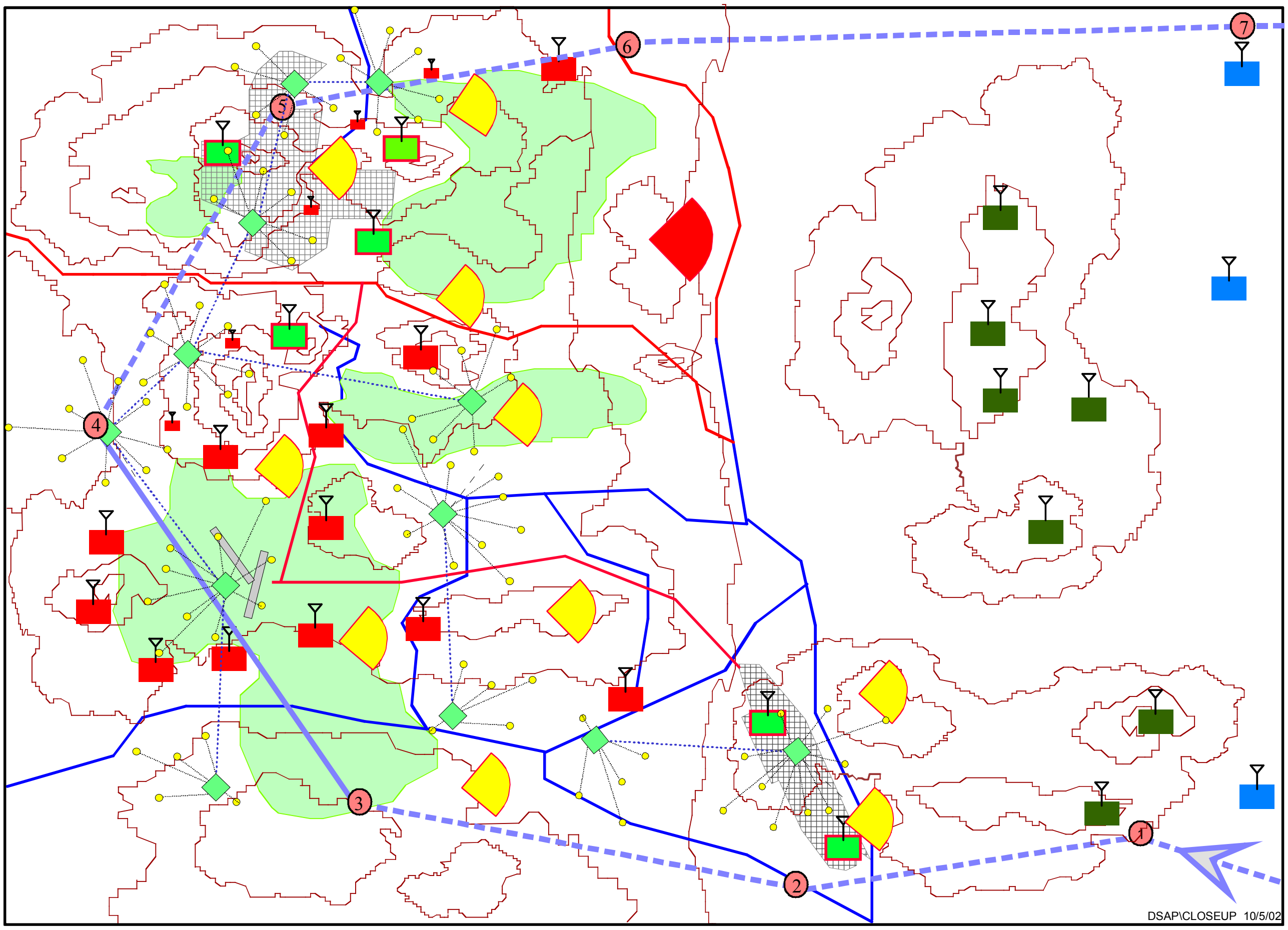

Figure 7-6. IADS simulation used to select missions.

Computation Of Event Probabilities

Within a mission trial, various events occur that are used to determine the probability of success. We will use simple examples to illustrate how the resulting sequence of probabilities can be used to determine a critical outcome. We must emphasize that these illustrations are greatly simplified to describe the types of probability calculations one can use. The methods described are not intended to represent the actual computational methods. We will start by considering the probability of a constraint violation.

Consider that a blue aircraft is attempting to avoid the red Integrated Air Defense System (IADS) get to a target. Let's suppose that jamming a red UHF link while flying a segment of the specified flight path through the IADS sufficiently reduces the probability of getting shot down, thus meeting the constraint. Looking at Figure 7-7A, the small ellipse, 1, represents the Elliptical Error Probability (EEP) around the position of the UHF antenna site represented by the dot. This implies that we know the position of the antenna within a probability distribution signified by the EEP.

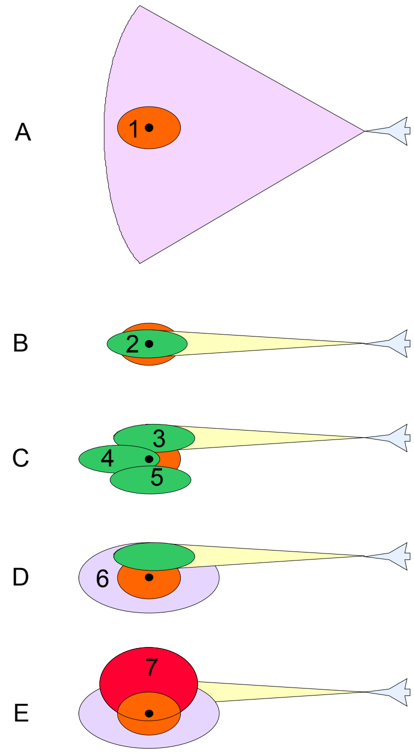

Figure 7-7. Computation of event probabilities within a sortie.

Now lets assume that the jammer has an effective area of coverage represented by the cone emanating from the aircraft. From the drawing, one can see that, even though we only know the position of the UHF antenna to within some error, the likelihood that we can jam it will still be close to 1 along a specified segment of our flight path. Even if the aircraft varies off its flight path by a small amount, or if the heading, pitch, or roll vary, it is likely we will still jam the UHF link during that segment.

The important consideration to be derived from this example is that one cannot simply add or multiply probabilities and get meaningful answers regarding the effectiveness of a system. This is not simply a case of accounting for many factors. It is the structure of the model that is most important. Generally, representing this information using linear models will produce results that are not nearly as accurate as those using nonlinear models. Furthermore, one must consider generating samples using the distributions or bounds to determine outcomes.

As another example consider Figure 7-7B where we plan to use kinetic ordnance on the target. Ellipse 2 represents the loci of locations of the point where the ordnance will hit, given that we aim at the center of the target. This ellipse represents an additional EEP of where the ordnance will impact the ground. This error is compounded in Figure 7-7C which shows additional ellipses, 3, 4, and 5, of possible impact locations relative to the worst case target locations based upon the original EEP of the target (ellipse 1).

Figure 7-7D shows the compound error ellipse, 6, as the locus of all possible points where the ordnance may hit relative to the actual target location. Ellipse 6 represents the distribution of points of impact accounting for the EEP in actual target location, and the EEP of impact about the estimated target location.

Figure 7-7E contains an ellipse, 7, showing the area of destruction centered at the point of impact. In the example shown, if the EEP perimeters are considered worst cases, then in the compound worst case condition, the target will still be destroyed.

There are numerous potential variations that must be analyzed carefully when computing the desired probabilities. For example, an error in heading for a long flight leg can cause error in position estimation. However, with GPS, this can be reduced. The pitch, yaw, and angles of the aircraft can change the coverage area of antenna patterns on the aircraft as well as the cross-sections presented to radar antennas on the ground. These effects are generally nonlinear, and must be dealt with as such. Else, the model error may be prohibitive.

From a modeling standpoint, this is a huge bookkeeping problem. The models are generally well known, and come in varying degrees of resolution. It is up to the modeler to determine the degree of resolution needed to produce valid results from a simulation.

Representing Sequences Of Events

When an aircraft flies multiple flight legs as indicated in Figure 7-6, it encounters sequences of events of the type described above. Each event can be conditioned on the prior event. The probability of violating a constraint will be conditioned upon multiple sequenced events. An example of how this may be calculated is provided below.

$$\Pi(\text{loss}) = 1 - \Pi(\text{success\_E\_n}|\text{success\_E\_n-1}) \times \Pi(\text{success\_E\_n-1}|\text{success\_E\_n-2}) \ldots \times \Pi(\text{success\_E\_2}|\text{success\_E\_1}) \times \Pi(\text{success\_E\_1})$$

A COA Of Multiple Sorties

At some point, the planner may decide to break the overall mission into separate sorties. For example, the first sortie may take out an air defense radar. This allows the second sortie to take out two SAM sites. This allows the third sortie to take out an electric power plant, etc. This can be described as follows.

$$\Pi(\text{success}) = \Pi(\text{success\_S\_n}|\text{success\_S\_n-1}) \times \Pi(\text{success\_S\_n-1}|\text{success\_S\_n-2}) \ldots \times \Pi(\text{success\_S\_2}|\text{success\_S\_1}) \times \Pi(\text{success\_S\_1})$$

This same concept also applies to the multiple mission case. We will address this in the following section. More importantly, this leads us to the next consideration, the reactions of the red force to blue actions.

RED REACTIONS TO BLUE ACTIONS

A significant aspect of the GSS IADS simulation is that red and blue force elements are represented by smart models. This allows blue actions to interact with red reactions and vice-versa. For example, blue aircraft can be equipped with self-screening jammers so that they neutralize the red radars. Sorties of multiple aircraft can contain escort jammers as well as ordnance to neutralize the IADS, allowing the principal weapon systems to get to the targets.

The Effects Of Red Reaction Time On Prediction Accuracy

If we are dealing with a single sortie, the reaction time will likely be in terms of hours or even minutes. Such short reaction times generally require preplanned approaches. The shorter the reaction time, the more autonomous the reaction is likely to be. The more autonomous the reaction, the more predictable it is likely to be. Therefore, the reactions to fast moving sorties are likely to be more predictable than higher level reactions to multiple sorties or multiple missions that span a longer time scale.

As an example, if we take out an air defense radar, it may be possible to assign a larger coverage area to another radar close to the area. It is likely that we can estimate the shortest reaction time to do this, and take it into account using the C2 and radar models. Similarly, if we take out a SAM site, it is likely that we can predict and model what the reaction will be during the course of a single sortie.

On the other hand, if we wait for a Battle Damage Assessment (BDA) report that may not be available until the next day, more people will enter the red decision process, and more assets may be brought to bear. We may then face a more varied set of much larger reactions. Because human decision chains are brought to bear with more options over time to mitigate the blue actions, the red reactions will likely be more difficult to predict.

There are two ways to combat this difficulty. We can model red's reactive decision processes relative to blue actions. The accuracy of prediction will depend upon the behavior of the leadership. Good leaders can be difficult to predict. Alternatively, we can plan sorties more carefully so that red has little time to react with a larger coordinated asset base. Just like the element of surprise, a major factor in neutralizing the use of defensive assets is the speed with which a sequence of sorties unfolds. As we increase the accuracy of predicting outcomes of sorties, we can better synchronize multiple sorties unfolding in parallel, as well as decrease the time between sorties. This gives the adversary little time to coordinate human decisions covering a larger asset base, relegating the assets to more autonomous behavior. This should serve to increase the overall prediction accuracy of mission outcomes.

Based upon the situation at hand, one must estimate the time frame within which one is dealing with autonomous behavior versus coordinated behavior with respect to significant asset changes. This determines the validity of current ISR information and error variances versus accounting for the human decision processes leading to significant departures from current ISR information. This can be couched in terms of discrete state changes as a function of time.

Time Frames Allowing For Significant Red Asset Redirection

We will now investigate what happens when the time between blue missions increases to the point that allows for red human decision processes to intervene. This allows for significant changes in assets and COAs on the part of red. Instead of dealing with reasonably accurate predictions of what blue is going to do to red, one must deal with potentially significant and unpredictable changes by red. This is particularly difficult given that red leadership will try to deceive blue, if not make it difficult to predict what they will do.

This is illustrated in Figure 7-8. The possible outcomes spread due to the time that red has to make changes. One could argue that the increase in variance may be nonlinear, increasing more with time. One can also argue that red changes will be constrained, limiting the variance. We are not attempting to determine the relationship between variation and time, only to say that as time increases, so does the variation. During these red reaction periods, ISR assets can be used to gain insights into who is communicating, who is moving, and what is happening.

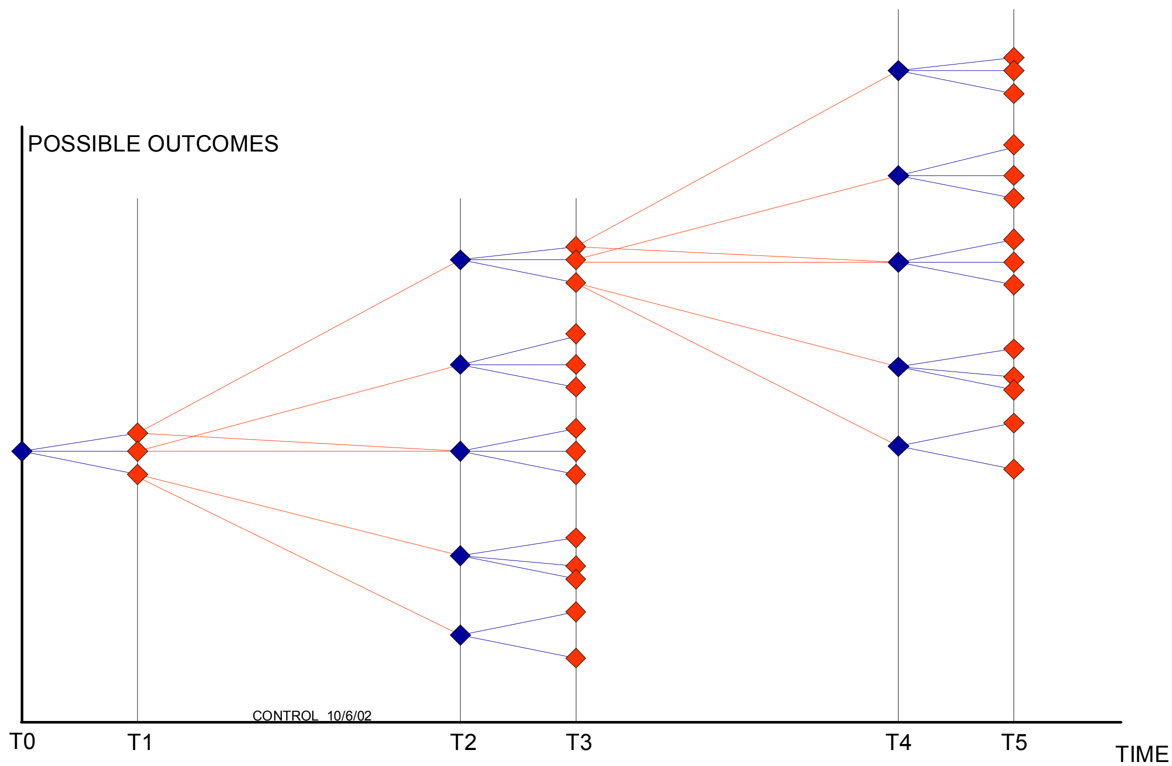

Figure 7-8. Computation of event probabilities within a mission.

Figure 7-9 provides an alternative view of the spread of possible outcomes when sequences of missions are packed into shorter time frames. The shorter the time period for red reactions, the smaller the variance from the current situation. This implies that as missions are packed tighter in time, including parallel operations when feasible, the more accurate the situation prediction will be for successive starting points.

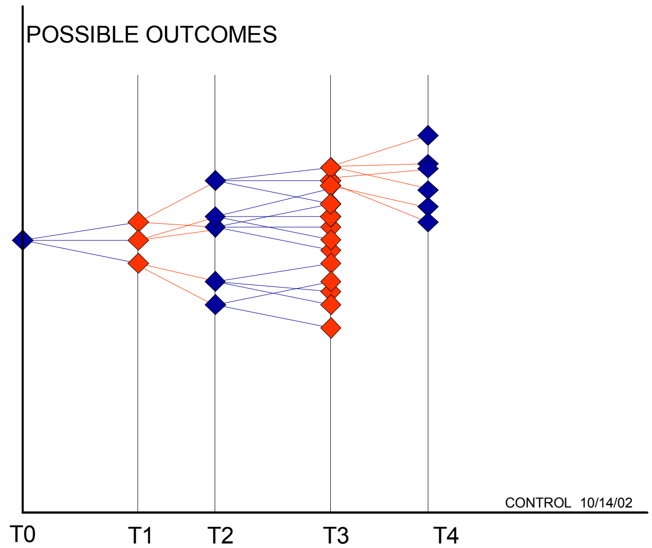

Figure 7-9. Computation of event probabilities within packed missions.

Packing And Synchronizing Multiple Missions Into Short Time-Frames

What does it take to pack and synchronize a large number of sorties into a short time frame? Clearly it takes more assets and more control over those assets. It also takes sufficiently accurate predictions of the end-states of sorties that affect the starting states of follow-on sorties. Based upon the probability calculations described in the above Section - A COA Of Multiple Sorties, we must deal with the effect of multiplying many probabilities. Each of the individual probabilities must be very high to keep the probability of success high.

$$\Pi(\text{success}) = \Pi(\text{success\_S\_n}|\text{success\_S\_n-1}) \times \Pi(\text{ success\_S\_n-1}|\text{success\_S\_n-2})$$

$$\ldots \quad \times\, \Pi(\text{ success\_S\_2}|\text{success\_S\_1}) \times \Pi(\text{success\_S\_1})$$

Given sufficient resources, backup or contingency plans can be laid to achieve higher individual probabilities of the individual sorties in the sequence. If sorties are independent, they can be accomplished in parallel, again requiring more assets. If they are not fully independent, it still may be possible to synchronize them, provided the degree of interaction can be predicted with sufficient accuracy. Finally, if the effects achieved can be assessed and communicated quickly, then one can make a decision to continue the fast pace, or to hold because of encountering unacceptable risk.

It is clear that, if blue can achieve sufficient surprise, speed, and synchronization, red's reactions to blue actions will be severely constrained. This leads to higher end-state prediction accuracy resulting from a set of blue sorties. Given sufficient accuracy of the end state predictions permits immediate blue follow on sorties. Given sufficient assets, control, and the ability to predict end-states accurately, speed increases the inertia that reduces red's ability to counter in an effective manner.

8. PLANNING PROCESS CONSIDERATIONS

FUSING FEASIBLE TRAJECTORIES

As used here, fusing feasible trajectories implies fusing feasible sub-COAs.

Military planning typically starts with a commander whose responsibility is to obtain a specified set of effects or objectives. An example is a JFC or the equivalent in classic battlespace, or in an asymmetric terrorist environment. In any case, actual planning and execution occurs across a coalition of organizations and sub-organizations that play a role in the process. Control is generally organized hierarchically.

A Typical Control Hierarchy

An example of a military control hierarchy is illustrated simplistically in Figure 8-1. At the lowest level of interest here, we encounter the need for a planning tool focused on the objective at that level.

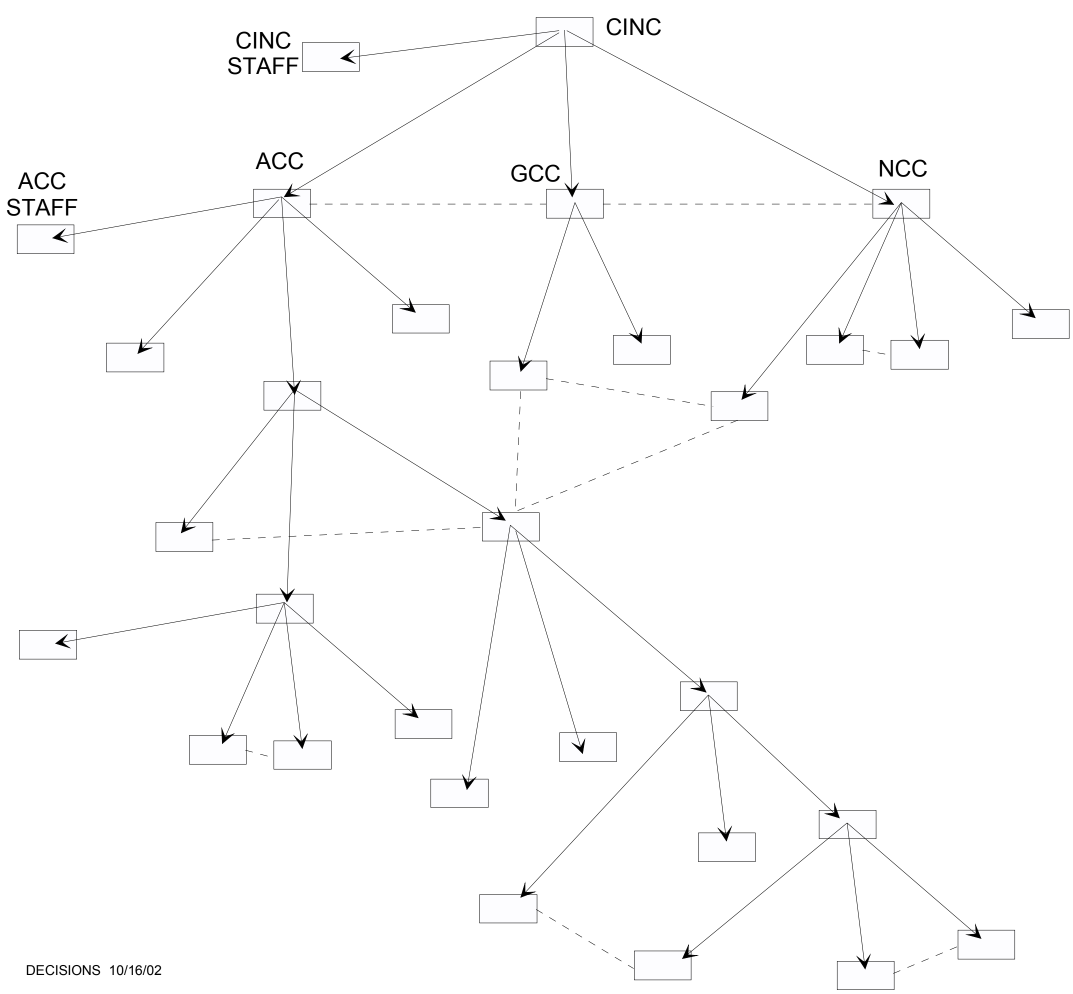

Figure 8-1. A simplified representation of a military planning process.

The Military Planning Problem - A Distributed Set Of Optimal Control Problems

Although control of a military force can be viewed as a classic complex control system, there are a few significant departures from the typical real time control problem. Probably most important is the distributed nature of the problem. The problem as the JFC sees it is just the tip of the iceberg. All of the boxes (cells) in Figure 8-1 can be considered control systems as shown in Figure 3-1, with each cell tasked to perform certain functions. For example, one of the cells may be a logistical planning group, tasked to produce a plan for getting troops from point A to point B within some constrained amount of time while minimizing some cost measure. Another might be an information operations cell, tasked to gather SIGINT. Another might be a cell in an Air Force squadron, planning a flight to take out certain sets of targets. Each of these must solve their own local optimal control problem. And each of these solutions can be affected by, and affect, some of its neighbors. Units must communicate plans, orders, ISR information, etc.

Information Exchange

Also shown in Figure 8-1 are communication cross links that provide for sharing of information across organizational structures as well up the chain of control. The product of an information operations cell can be input to the cell planning missions for specified targets as well as for many other boxes in Figure 8-1. Output of one cell can be observable inputs to other distributed control elements. When analyzing a control system, part of the problem is the accuracy of the information inputs. Variations in the data will cause variations in the predicted responses that are used to determine the optimal control sequences. Furthermore, getting the right information to the right spot within a useful time frame is a significant part of the problem. Because of the distributed nature of the controls, understanding these information exchange requirements is a major consideration.

Fusing The Trajectories

There are two important factors to emphasize. First, the individual cells are like the salesmen who have considerably more information upon which to condition the envelopes of their trajectories of critical events in time. Second, these trajectories and envelopes can feed the next layer up, or possibly a peer layer, to improve the conditioning on the probability statement for their trajectory. This is illustrated in Figure 8-2, where the information flows are shown, independent of whether they are following the control hierarchy or are peer interchanges.

The current approach to formulating and interchanging this information in the AOC is somewhat informal. An appropriate set of tools as depicted in Figure 8-3 can improve the means for interchange as well as generation of the critical information required to maximize accuracy of prediction at each layer up the hierarchy.

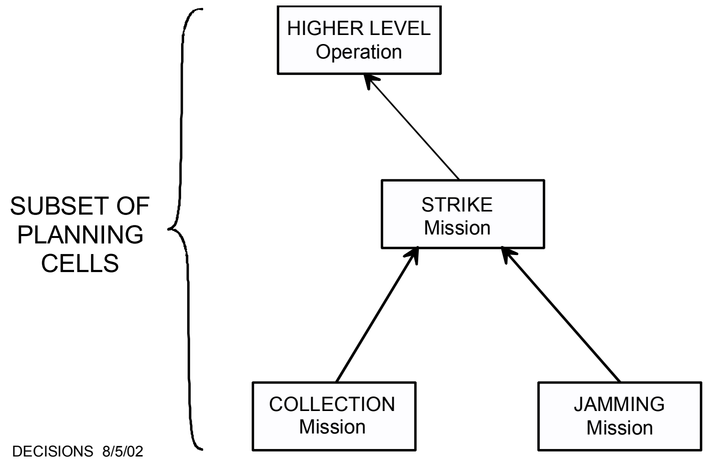

Figure 8-2. A simplified representation of a military planning process.

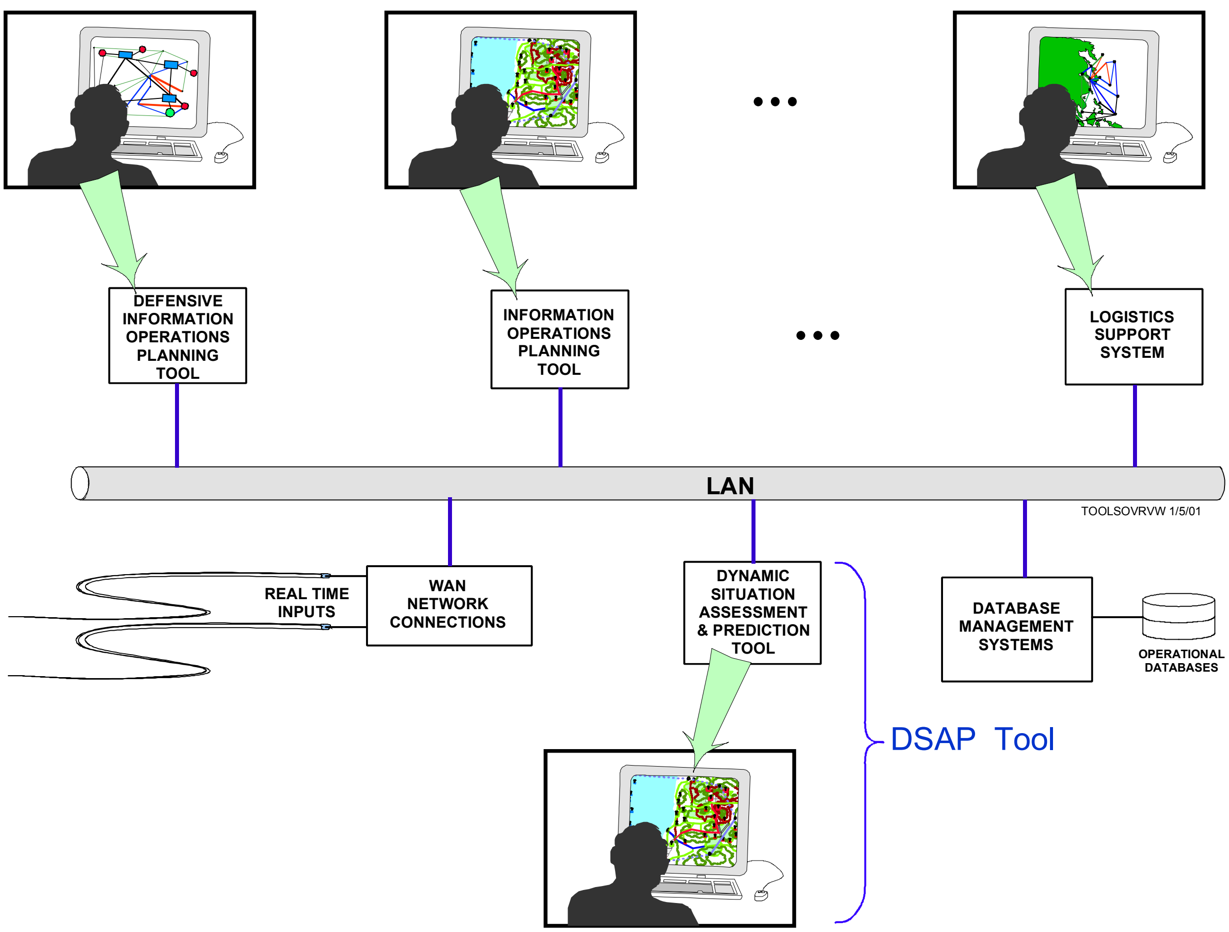

Figure 8-3. An integrated set of DSAP tools.

Referring back to Section 3, Figure 3-2 provides a detailed block diagram of a typical control system containing an imbedded prediction system. This is redrawn for simplicity in Figure 8-4a below. In the case of a distributed control system as depicted in Figures 8-1 and 8-2, each element takes on the form shown in Figure 8-4b below.

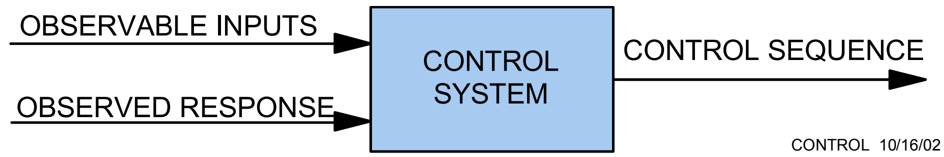

Figure 8-4a. A typical control system.

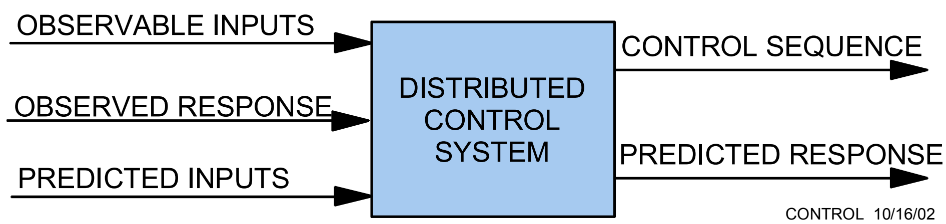

Figure 8-4b. A distributed control system.

The difference between Figures 8-4a and b above is the explicit output of predictions from one tool for input to another tool. Each tool that provides predictions to another tool must also provide the characterization of prediction error to the receiving tool. This error characterization must be used by the receiving tool to determine the error of its own predicted response. The error from the predicted input must be used to determine the feasibility of meeting constraints, as well as the optimized control sequence.

Characterizing Predicted Or Forecasted Responses

Figure 8-4b takes in predicted responses and produces predicted responses. This will not always be the case. Many tools do not produce predictions, including error characterizations and confidence levels, as defined here. The types of responses produced by various tools of interest must be characterized. For example tools that use engineering models may have the error characterized based upon prior testing and analysis. Sensors typically provide CEPs and EEPs as indicated in Section 7. These can be used as predictions provided that the environment in which the model and real equipment are used does not nullify the validity of results. Models that produce forecasts are addressed in the Section 10.

REUSING PLAN SEGMENTS

When planning missions under severe time constraints, it is most desirable to be able to draw on libraries of missions or other forms of plan segments so that most of the details are already built. The trade-off between starting with a segment that must be changed and starting with a clean slate depends upon the time it takes to get the job done. Most often it should help to start with a plan that has been used in the past.

State Sequences That Appear Repetitive

In general, we cannot say that the sequence of events in a competitive engagement involving large numbers of entities will be repetitive or cyclical. Repetitions may appear to occur, but they are most likely different to a small but important degree. This is to be expected since adversaries will look to take advantage of what appears to be a repeated direction to invoke surprise.

This is not to say that subsets of a plan may not be viewed as repetitive parts. However, the context is likely to be significantly different. Using Figure 8-5 as an example, going from A to B at time TL may look the same as going from F to G at time TM. However, this is different from saying that states repeat in time. Time itself is part of the system state vector, and anything that depends upon time may make the state transition different, even if we remove time from the state vector explicitly.

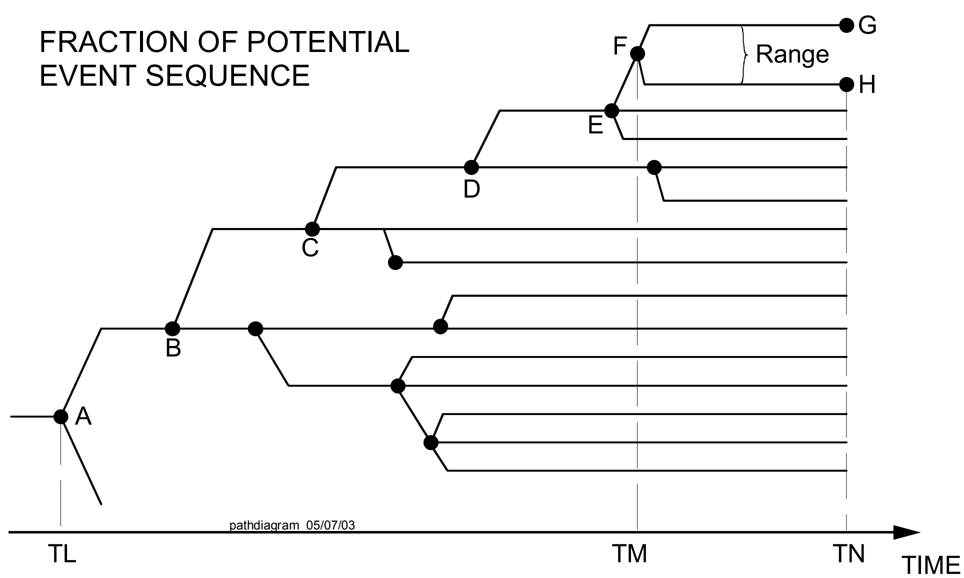

Figure 8-5. A fraction of a potential event sequence in a planning tree.

If subsets of an event sequence can be treated as *independent* relative to their context, then they may be reused. This is explored in the next section.

Properties Of Reusable Plan Segments

Can parts of a plan be reused? If they can, they may need some tailoring. The important property is that these segments are sufficiently independent of their context or environment. The more independent, the less tailoring required. The measure of independence is determined by the initial and terminal states, and the internal state of the system at the initial time. Finding opportunities for reuse may be very productive. At some point, tailoring becomes less productive than starting with a clean slate.

APPROACHES TO REPRESENTING END STATE PROBABILITIES

Finite State Machine Representations

Techniques based upon finite state machine representations, e.g., those dealing with event trees, are known to have difficulty with scalability. For example, the tree segment in Figure 8-5 may be a very small fraction of a real state diagram as a function of time. Note that, starting at time TM, there is a single state - A. At time TN, there are 16 states spawned from state A. It is not hard for a realistic plan to evolve potentially to thousands of states. The difficulty of calculating the end state probability grows rapidly as the number of states increase. What's more, this approach appears unsuitable when dealing with states that take on continuous values. This is discussed below.

A finite state machine can be used to represent elements of a digital computer, particularly the arithmetic operations, e.g., add, subtract, multiply, and divide. However, to represent a small program using even a small memory can be intractable. If one or more states in a program are represented by real numbers, representing them directly using a finite state machine becomes effectively intractable. *The underlying problem is that of selecting the most convenient coordinate system to represent the dynamics of the system of interest.*

Representing States With Infinite Outcomes

The finite state machine approach assumes that all states have a finite set (typically a relatively small set) of outcomes. In many real planning situations there are cases that must be represented by continuous outcomes for accuracy. When an attribute is represented by a *real* number, e.g., a probability between 0 and 1, or any continuous variable (e.g., temperature, spatial coordinate, etc.), there is an infinite set of outcomes or states.

As an example, suppose that F in Figure 8-5 can advance to a continuous range of outcomes between G and H. This is very easy for a digital computer to represent. One typically uses 32 or 64 bits to represent a real number. That's how nonlinear differential equations get solved using computers. Most engineering problems deal with real numbers; they could not be limited to integers or finite states. It is very difficult, if not intractable or impossible, for methods that depend upon a finite state machine representation to deal with such problems.

Representing Distributions Using Stochastic Models

Solution of the infinite state problem is common in engineering. One is looking to derive sufficiently accurate estimates of the values of continuous functions, or determine distributions representing the components of the end state response vector, Z. (Z is derived from the state vector, X, which represents the dynamic system.) This does not require an exhaustive solution for all the intermediate states. When dealing with real numbers, this would require a space of multiple orders of infinity.

To estimate end state distributions, one typically creates a *stochastic* state space model that represents the transitions of state. It is not unusual for models to contain millions of simple rules governing transitions of huge state spaces. The stochastic version of the generalized state space model can be based upon the distributions at each transition point, [20], [21], [22]. To investigate worst cases as defined in Chapter 6, one need not know the precise shape of the distributions, only a set of bounds that encompasses the possible outcomes.

By using simulation, one can select samples from these distributions to generate the resulting path from the initial state to the final state. There are various ways to generate enough paths to provide a sufficiently accurate estimate of the distribution of the terminal state. Monte Carlo is a popular approach. It is implemented by taking random samples from the distributions at each transition, or at any point where the possible outcomes affecting the final state are represented by a distribution.

By virtue of the central limit theorem of statistics [24] - and given that: (1) the bounds on the transition distributions are represented with sufficient accuracy, and (2) a sufficient number of sample paths are generated - the resulting end state distributions should converge to those of the real system. Generating a sufficient number of paths is relatively easy. We will discuss the requirement on bounds of the distributions below.

One of the significant differences between this approach and those using a finite state machine is the recognition that one is looking to characterize a distribution of the end state. This is required in order to provide confidence measures for the probability statements. Given that we must produce an estimate of the distributions, we can accomplish this via statistical sampling. In fact, the models used to generate the samples can be very accurate since we are only computing a single sample path and the number of sample paths required is likely to be quite small. Most likely less than 100 paths will have to be generated. Therefore we can focus on model accuracy instead of looking for ways to rapidly run through all possible paths of a much less accurate model.

Representing Bounds On Distributions Using Stochastic Models

As described in Chapter 6 and the corresponding references, one typically takes a worst case approach when the distributions needed to describe the variations are unavailable. This can be done with minimal data, or by making judgments about reasonable limits. Multiple powers of ten rarely apply to ranges of numbers, and when they do, someone typically knows reasonable ranges, or one can come up with limits that are imposed by obvious logical or physical limitations. Important parameter limits can be researched. One can then perform parametric and sensitivity analysis and draw conclusions based upon simulation results.

Predictions using probability statements

By definition, probabilities are derived from distributions. Without sufficient knowledge of the distribution of values that an attribute takes on, one cannot determine the probability that a particular value will be exceeded with any consistency. This can be achieved without knowing the shape of the distribution provided one has boundary values to determine probabilities (unknown but bounded distributions [19]). In any case, one must provide a confidence in the probability statement, or it is not very useful. If one analyzes how the probabilities are derived, and it turns out that there is less than 50% confidence in the probability statement, this must be factored into the use of the probability statement. Otherwise, when one goes back and looks at the outcomes, they may not reflect the apriori probability statements. When one has insufficient data to formulate all of the distributions, unknown but bounded distributions can be used with a worst case design approach.

Posing the worst case design problem

The worst case design problem in engineering provides an excellent example of the solution to prediction problems. See References [13] through [18]. Consider that the Test Sample trajectory inside the shaded area in Figure 2 must satisfy two constraints selected from operational requirements. First, during the initial (TI) to final (TF) time period (TI, TF), the trajectory must remain above the H(1) = 0 line. Second, it must remain below the H(2) = 0 line. These two constraints are expressed as inequalities.

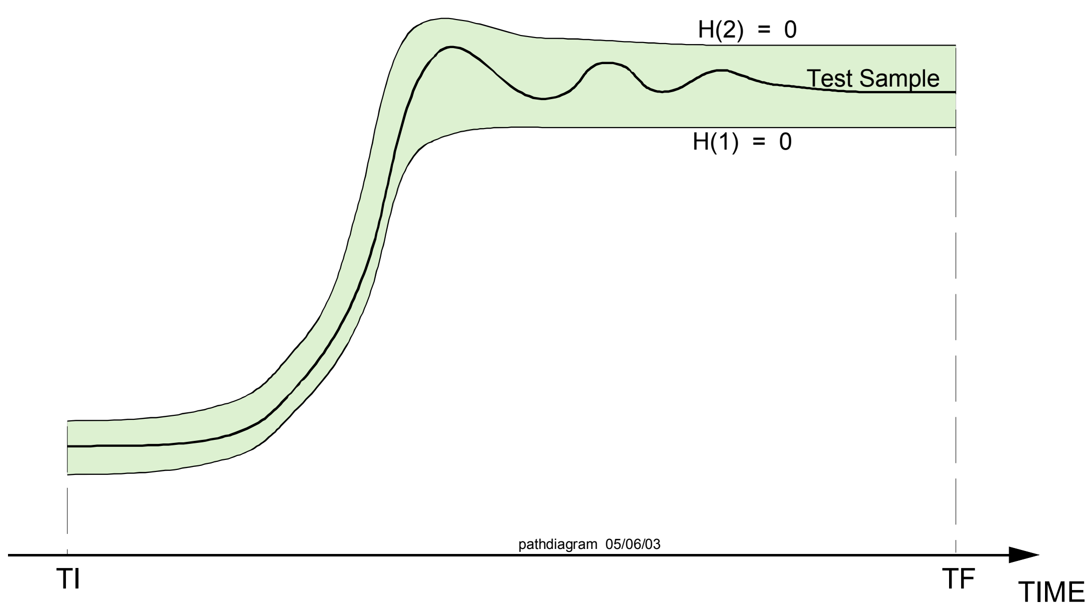

Figure 8-6. A worst case design problem.

In a typical worst case design problem, the two constraints must be satisfied for all conditions reflecting the range of possible parameter variations one could encounter over the test sample trajectory. The nominal design is usually centered within the constraint boundaries. The worst cases occur when the trajectory comes closest to either of the constraint boundaries. Additional variations may violate the design constraints. Typically, the constraint boundaries can be tested independently to find a worst case condition for each constraint.

In addition to meeting the design constraints one also wants to optimize the design. For example, one may want to minimize power or minimize state transition time while meeting the constraints. In general, one is usually trying to minimize energy spent, resources used, or time to achieve a goal. This typically pushes the trajectory to the worst case limits.

Worst case optimal design problems of the type described above have been solved for problems with large numbers of constraints and hundreds of transition equations. The optimization algorithms contained in GSS have been proven to work well for these types of problems

POSING THE WORST CASE OPTIMAL CONTROLLER DESIGN PROBLEM

This section explores the elements required to define the worst case optimal control problem. The intent is to understand what is required to deduce a well defined problem when faced with variations in a nominal trajectory that can cause constraints to be violated. In addition, we are concerned with controlling a system that is changing in real time. Thus, we are predicting a sequence of optimal controls to be used up to some terminal state, watching what happens, and then updating the optimal control sequence starting from the current state.

Initial and terminal states

To properly define the optimal control problem, one must have well defined initial and terminal states (TI and TF in Figure 8-7). These states may be parameterized, but the ranges must be defined as a set for which someone will provide the numbers.

At a higher level, one may have to solve multiple optimal control problems to solve the overall problem including the identification of worst cases for each constraint, and optimization of selected criteria. Worst case constraints may have to be met at initial and terminal states that must match given requirements. For example, if the terminal state of one subsystem is the initial state for a subsequent subsystem, the worst case outputs must fall within acceptable limits of the worst case inputs to the follow-on system.

Isolation of subsets of the planning problem

One has to determine the time frame of a scenario, i.e., the values for TI and TF, beyond which the planning process changes too much, variations, grow too large, or plans are just not of interest. Given a sufficiently accurate model including the bounds on variations over TI and TF, one must know the initial conditions at TI with sufficient accuracy. If simulations take many minutes of time, it may be possible to update state estimates in real time. If Monte Carlo or optimization is used, then TI may be updated in real time.

In the field of circuit design, CAD researchers tried to automate the approach to architecture, i.e., what is the best way to structure the overall network. But experienced circuit designers could generate innovative architectures rather quickly. One could perceive automating the selection of predefined architectures, but even that had substantial difficulties, and could not appear to be at all competitive with human experience.

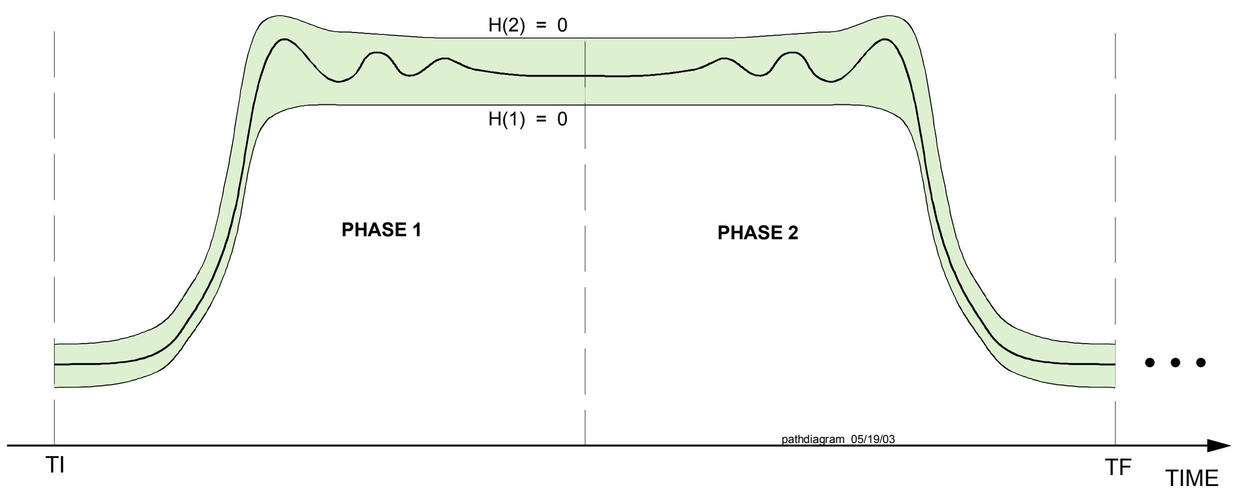

Figure 8-7. Matching the initial state of the next phase with the terminal state of the last.

Using A Scheduling Approach Versus A Simulation Approach With Optimization.

Both the scheduling approach and simulation approach use optimization algorithms. Scheduling algorithms generally use linear programming methods. The problem is accounting for the nonlinear nonstationary effects that determine realistic model behavior. They can be used in tandem as shown in Figure 8-8, with a man-in-the-loop to do the final optimization.

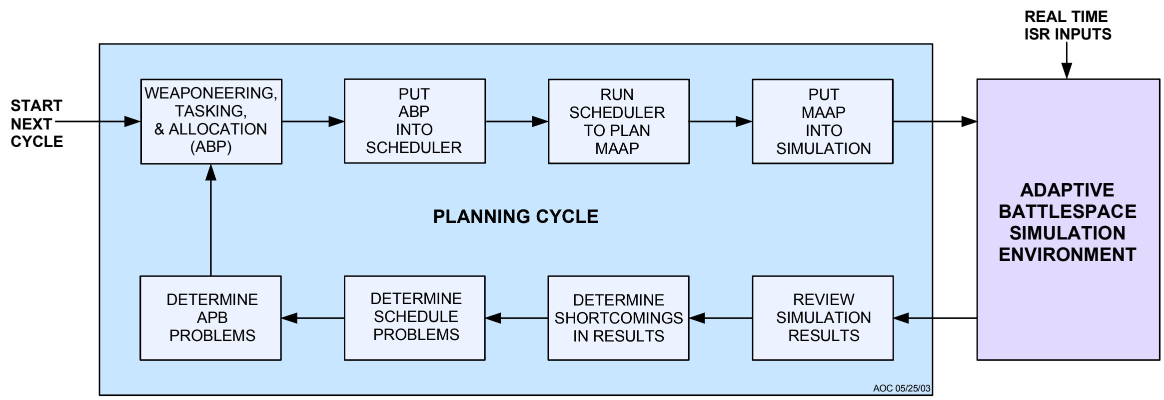

Figure 8-8. Using a detailed battlespace simulation to predict the outcome of a plan.

Alternatively, if the simulation runs fast enough, and enough smarts are built into the models, the battle space simulation could be presented with the desired sequence of steps and built-in optimization could be run to determine the schedule with all worst case conditions applied automatically. This is the worst-case stochastic optimal control problem.

One can conceive of presenting the desired end state description to the battlespace simulation and having the sequence of steps determined automatically. At this point in the process of developing and integrating technology for solving the overall planning problem, efforts are best focused upon using human intervention to develop the desired sequence of events with specialized tools to perform this function.

Using smart models to represent adversarial reactions

The battlespace simulation must represent red reactions to blue actions. For example, if blue aircraft fly over red airspace, they are detected by red radars. Track information is passed to red C2 centers to be processed. The C2 centers assign weapon systems to track and engage blue aircraft.

In general, one can use smart models to represent both friendly and adversary reactions to the others actions. To do this, the appropriate adversarial model state vector sequences, and transition probabilities required must be available.

9. PLANNING FOR THE WORST CASE

In Section 7, we imposed the constraints that each aircraft complete its part of the mission and return to its base - intact. This could also be imposed using small but finite probabilities that a loss could occur, and lumping them in with the optimization criteria using a very large multiplier to weight them heavily. This second approach is easier if one is constrained to the use of linear mathematical-statistical models. However, the desired constraints are not separated out, and the large multipliers tend to blur the real boundaries. Working with hard constraints that must be met in order for solutions to be feasible helps to align the definition of the problem with real world requirements. However, the problem becomes much harder to solve without special algorithms to find feasible solutions, and optimal solutions that meet all of the constraints.

Given a general discrete-event modeling environment, where nonlinear actions and reactions can be modeled in detail, both the definitions and measures of satisfaction of constraints can be described quite accurately. One can determine the probability of failure in terms of a constraint boundary, and the probability of failure if that boundary is crossed. Good judgment can be used to define these constraint boundaries in a precise pass/fail manner. This makes it clear when constraints have been violated.

When characterizing red reactions to blue actions, one must deal with the worst case scenario. This can help to simplify the problem. For example, if there are a set or range of reactions that can be taken, one simply analyzes the reaction that is worst-case. As indicated in Chapter 6, worst case must be considered independently for each constraint. It occurs when that constraint goes maximally negative (independent random variables are selected to maximize the probability of a constraint violation). Thus the worst case for a given constraint can be determined by finding the combination of reactions that maximizes the probability of violating that constraint. This can be accomplished automatically using nonlinear optimization techniques as described in references [14] and [17].

10. Fusing Forecasts With Predictions

Not all of the cells that have planning tools will be able to provide predictions. Some will only be able to provide forecasts. Figure 10-1 illustrates this situation. One is faced with the question of how to gain predictions from forecasts.

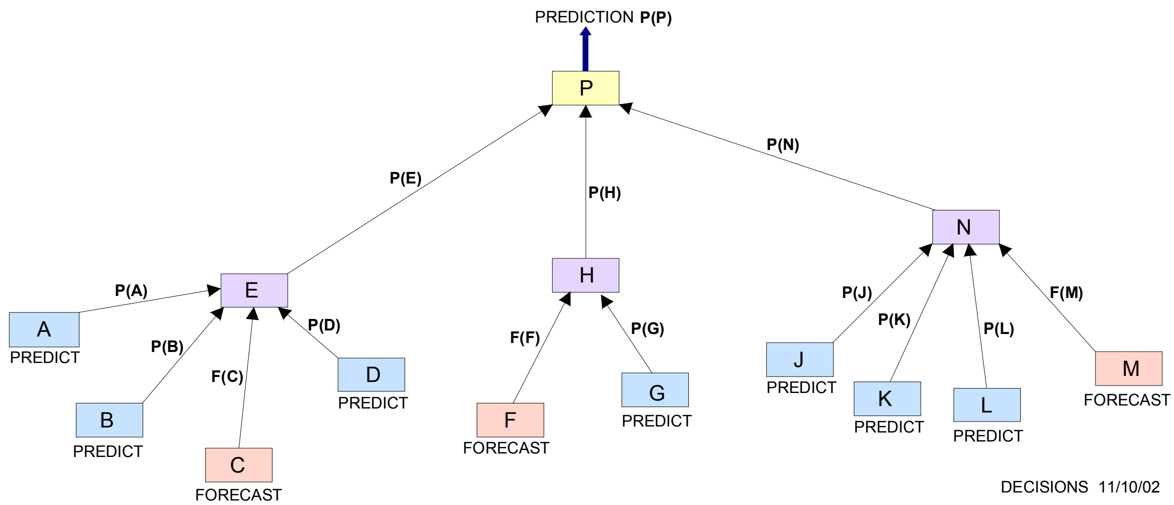

Figure 10-1. Fusing forecasts with predictions.

This presents the question: How can we fuse forecasts with predictions? The solution to this problem can be approached in two ways. One way is to plug in worse case outcomes or trajectories in place of the forecasts and determine if feasible solutions can be found that are acceptable. Another approach is to perform parametric and sensitivity analysis to determine the range of possible outcomes. These are described below. In either case, one is trying to derive statements about future outcomes. Once a forecasting method generates sufficient predictions that can be compared to real world outcomes, a prediction can be developed using the measured error data.

Fusing Worst Cases Forecasts

This approach requires analysis to determine what the worst case outcomes are. In general, outcomes are characterized by a *state vector* of attributes as discussed in Section 4. The reader is also referred to Section 6 describing the worst case design problem. Before a forecast or prediction can be made, the set of possible outcomes (outcome states) must be enumerated. This may involve range limits on real numbers, a set of states that a status variable can take on, etc. If one is not able to make a prediction, one can determine the worst case outcome and use that as an input to the next tool. If this worst case input is the only one, with many prediction inputs to the next tool, the resulting prediction from that tool may yield satisfactory results. If the worst case is too severe of an input, the output from the next tool may be unsatisfactory.

Parametric And Sensitivity Analysis

If the worst case approach presents conditions that are too severe or costly, then one can perform parametric and sensitivity analysis to relax the conditions. This requires performing large numbers of simulations to characterize the probabilities of outcomes in terms of variations of the input parameters. If one can assign probabilities to the inputs, then one can make a prediction. If not, a more accurate forecast can be used as a proxy for a prediction.

11. SIMULATION ARCHITECTURE CONSIDERATIONS

Starting with the simulations required to support the mission planning process shown in Figure 7-3, we will examine the model architectures from a more generic perspective. Figure 7-3 indicated the need for at least five simulations. These were made up of the IADS simulation supported by four other simulations. This was merely an illustration of the approach to using simulation to support the mission planning process. We will now take a more generic look at how models and simulation architectures might be organized to support more complex realities of this planning process.

Types Of Entities To Be Modeled In A Battlespace Simulation

In the North East Asian (Korean) scenario for the JTIDS network management effort, four naval battle groups are modeled along with friendly ground forces, including air defense systems, as well as all of the air assets including sensors and C2 systems. The types of entities to be modeled in the active simulation may include, but not be limited to, the models listed below.

Targets - Targets of opportunity for blue forces, specified objective targets, immediate targets, etc.

Blue Aircraft - Reconnaissance planes, fighters, bombers, etc., with dynamic changes to organizational and operational assignments, detailed flight plans, refueling, targeting, etc.

Blue Naval Vessels - Carrier battle groups, including the carriers, AEGIS, destroyers, submarines, etc.

Blue Ground Vehicles - Tanks, Bradley fighting vehicles, armored personnel carriers, etc.

Blue Sensors - Including radars and other detection systems.

Blue Jammers - Including radar and communications jamming systems.

Blue Weapon Systems - Including the closed loop message traffic that must be concluded in order for a weapon to be successfully set on a target. Includes probabilities of kill.

Blue Special Ops - Including ground forces.

Blue C2 Nodes - Includes reception of messages calling for weapons on targets, decisions of what weapons are best suited to what targets, and assignments of targets to particular weapons.

Blue Communications Traffic - INTEL Tracks, Locations, etc., C2 voice and data messages, and other traffic.

Blue Communications Systems - Ground wired and wireless systems, satellites, etc.

Electro-Magnetic Environment - Effects of terrain, foliage, atmosphere, correlated and uncorrelated noise, etc.

Red Aircraft - Reconnaissance planes, fighters, bombers, etc., with dynamic changes to organizational and operational assignments, detailed flight plans, refueling, targeting, etc.

Red Ground Vehicles - Tanks, other fighting vehicles, armored personnel carriers, etc.

Red Jammers - Emission of different types of waveforms over different parts of the spectrum.

Red Air Defense Radars - Scanning, emission, and detection to determine if friendly platforms are being tracked.

Red Weapon Systems - Includes the message traffic that must be received in order for a weapon to be successfully set on a target. Includes probabilities of kill.

Red C2 Nodes - Can include reception of messages calling for weapons on targets, decisions of what weapons are best suited to what targets, and assignments of targets to particular weapons.

Red Communications Traffic - INTEL Tracks, Locations, etc., C2 voice and data messages, and other traffic.

Red Communications Systems - Ground wired and wireless systems, satellites, etc.

MODEL AND SIMULATION HIERARCHIES

Figure 7-3 illustrated a simulation hierarchy consisting of an IADS simulation being fed by four other simulations, two of which (TEL-SCOPE and SAT-COMMS) were communications simulations. These examples were used to reflect the most recent use of simulation to support the planning process for existing Air Force applications. We will attempt to expand on this concept to achieve a more generic view of model and simulation sizes and hierarchies.

Requirements For Modeling Large Numbers Of Entities

Figure 11-1 illustrates potential model hierarchies that could be used to support a variety of planning tool applications. Again we are limiting ourselves to the IADS type application. At the top of the figure are the basic models required in the IADS simulation. To simulate a single mission with sufficient accuracy, one must model the ISR capabilities, the C2 effects, and the weapon systems for both red and blue to a required level of detail. This implies modeling specific equipment available to perform for that mission, including the platforms, communications equipment, jammers, and ordnance pertaining to red and blue. One must model blue fuel availability and the refueling part of the scenario when multiple sorties are required.

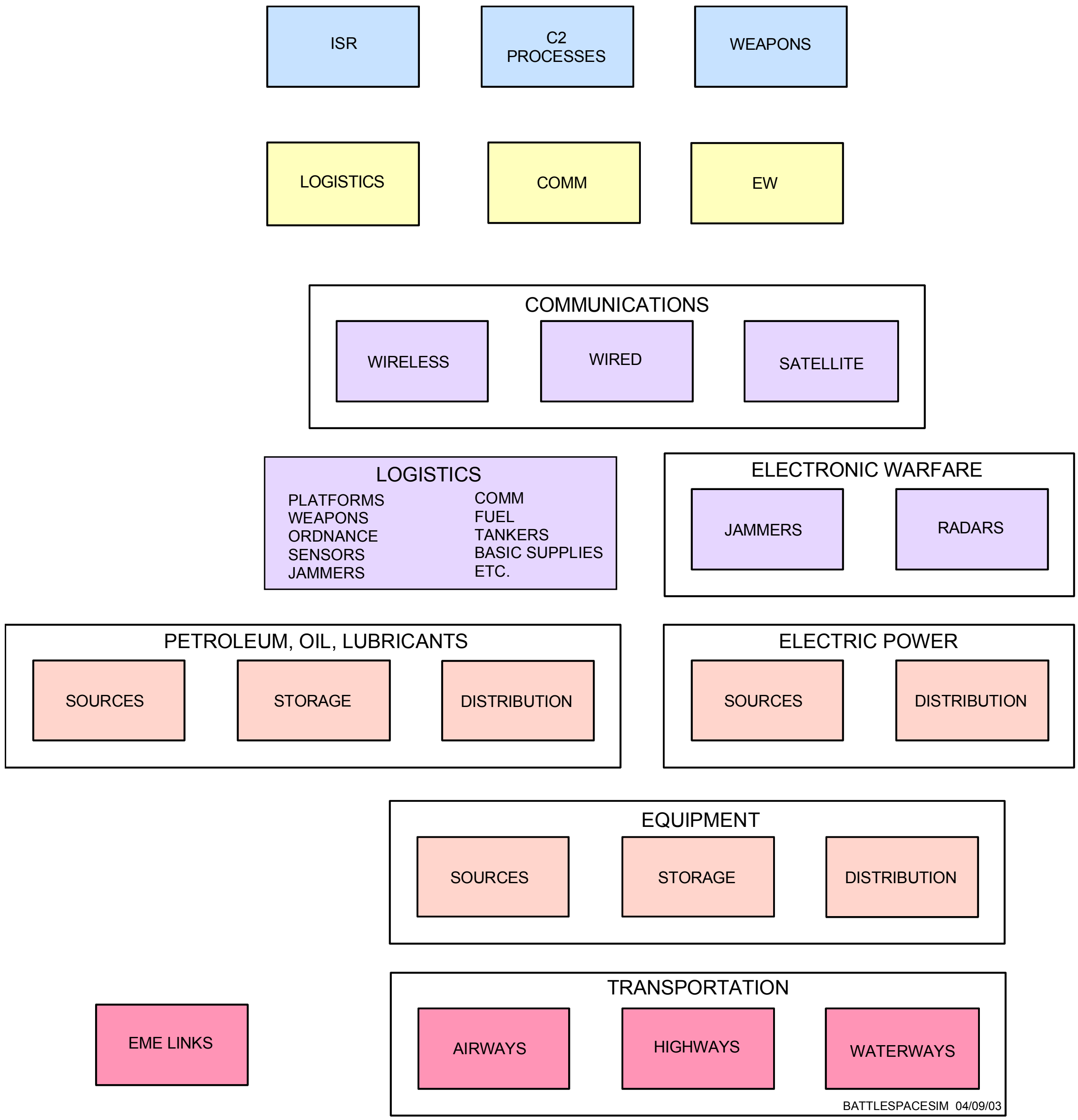

Figure 11-1. Illustration of hierarchical model architectures.

Requirements For Multiple Resolutions

The different colors are used to represent different levels of resolution. The first level models - blue boxes - must model the interactions of ISR, CE, and weapon systems. These elements must communicate using a set of communications models - yellow box. These comm models may be at a relatively low level of resolution, indicating the probability that messages will get from A to B.

The higher level resolution communications models (in purple) are needed to compute the probability of communications for the lower resolution models. These must be updated by the EME LINK models (darker red) when units change power levels (turn on and off) or move, or when jammers move or change power levels in the band of concern. These higher level models can be contained in separate simulations. This provides independence, affording ease of maintenance and the ability to support multiple simulations. It also provides potential speed improvements if run on separate processors.

Requirements For Subject Area Experts

As indicated a number of times above, human judgment will not be eliminated for the foreseeable future. Subject area experts will be called upon to make the decisions, using the best tools and automation available.

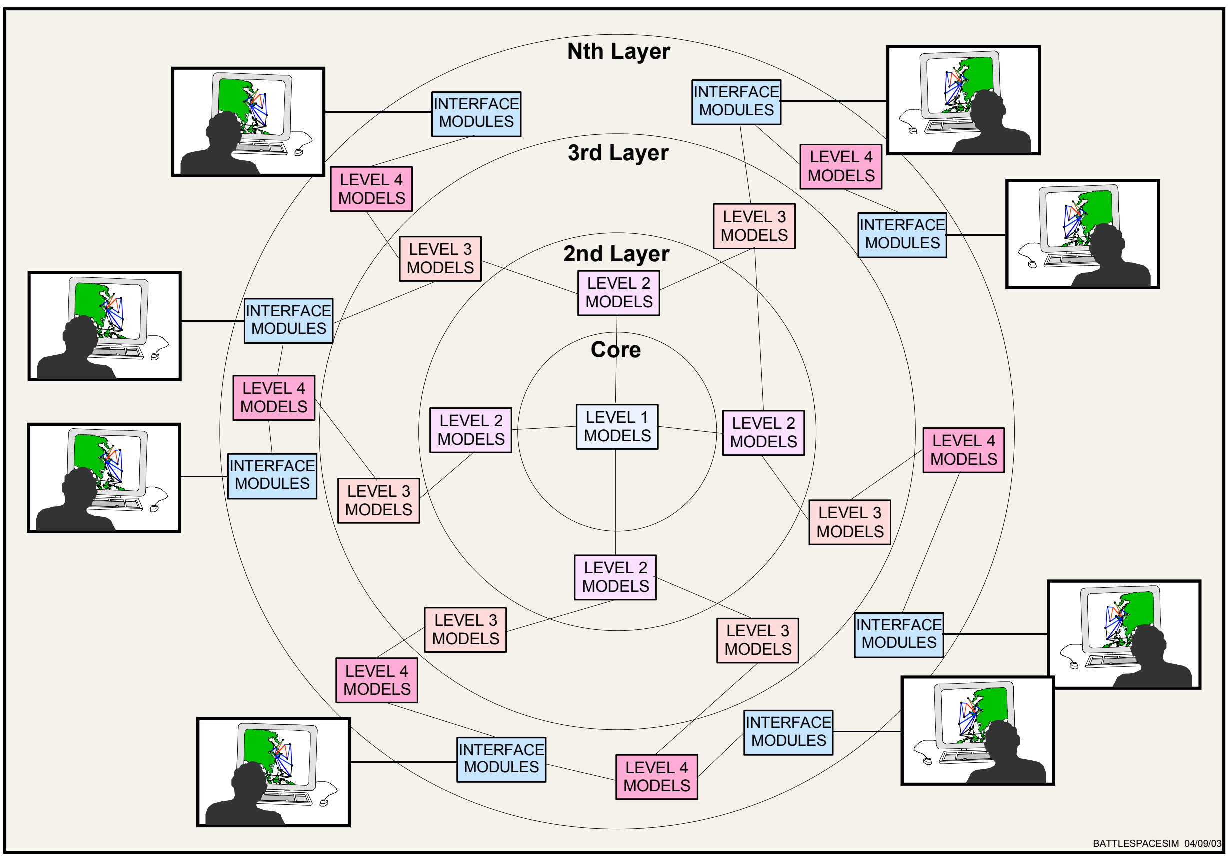

Figure 11-2. Illustration of hierarchical model architectures with subject area experts.

Requirements For Graphical Displays And Background Overlays

Visualization of scenarios for military planning generally starts with a map. If a reasonably detailed map is not available, military planners will not use the tools. Maps must reflect the scenario. Figure 11-3 illustrates a graphical display with a map background.

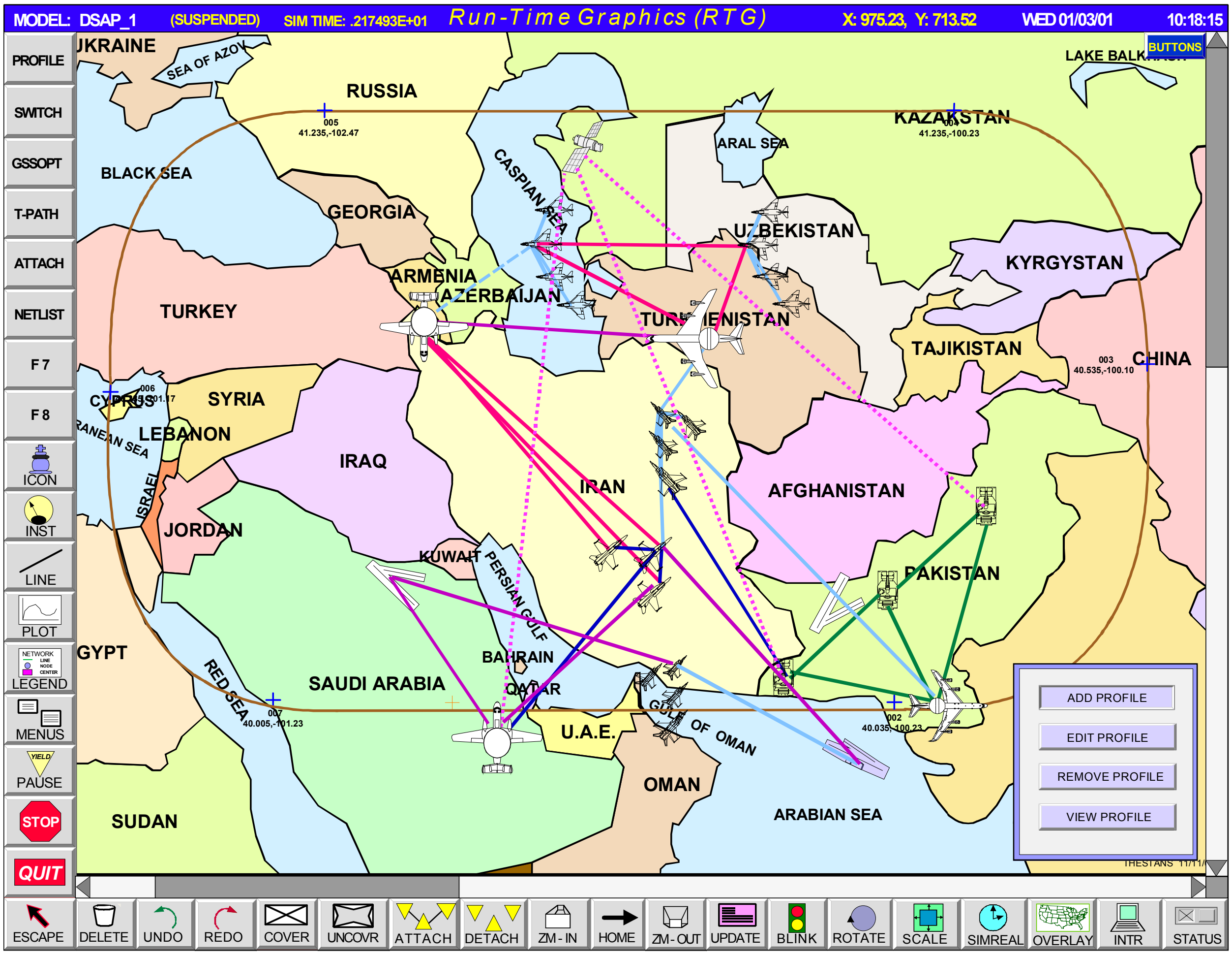

Figure 11-3. Illustration of graphical display of selected operations.

Fully flexible electronic maps allow the viewer to pan and zoom over very large land areas. In most areas of the world this is accomplished using 6º by 8º UTM grid zones that have a width of approximately 600 to 700 Km and a height of 800 to 900 Km. Each grid zone provides a separate Cartesian coordinate system for fast x, y calculations. These grid zones are fitted together to form the map. An illustration is shown in Figure 11-4.

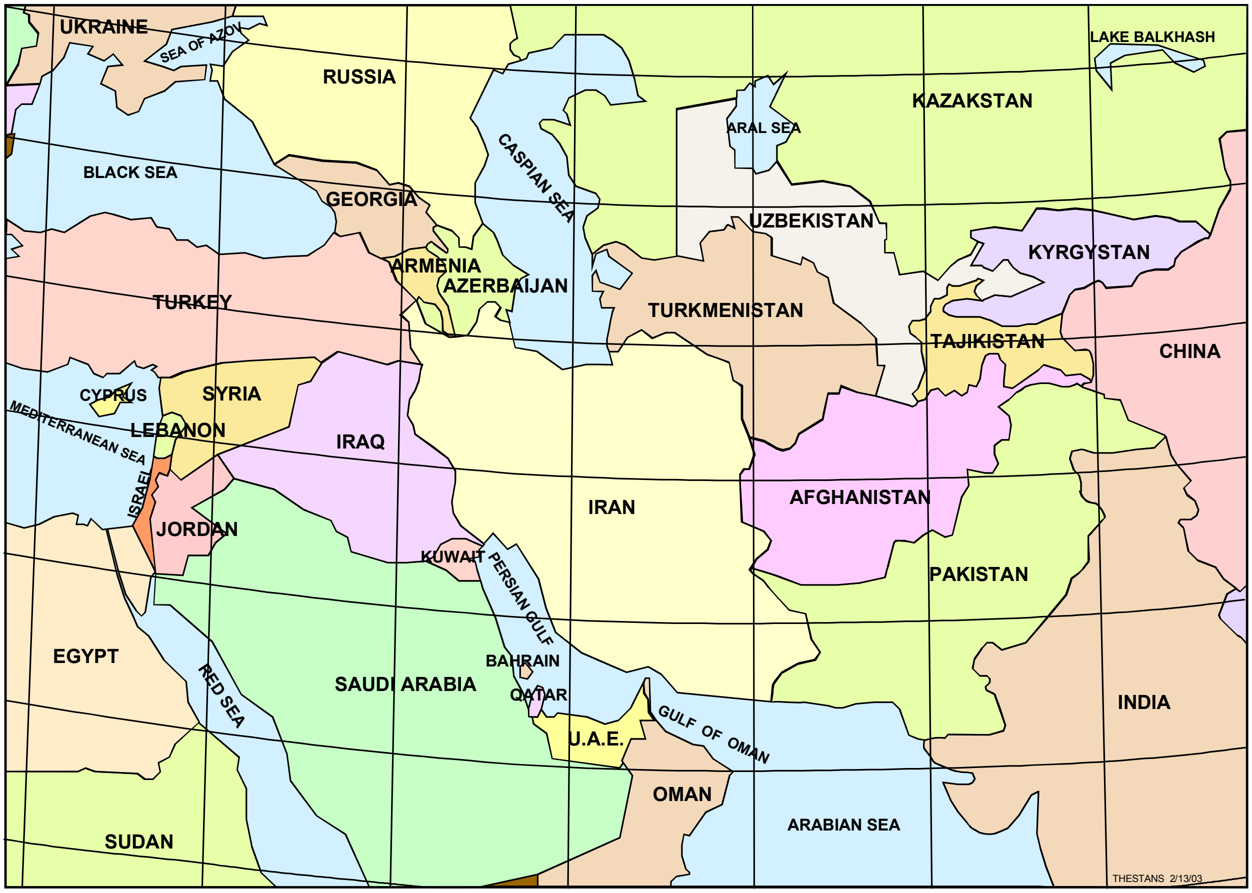

Figure 11-4. Illustration of the size of terrain area required to support a Mid-East scenario.

The lines of longitude and latitude shown in Figure 11-4 are rough approximations of grid zone boundaries, used principally to indicate coverage. Since the grid zone boundaries are fixed, a land area width of 700Km may require 3 grid zones of latitude. Similarly, a land area of 1000x1000 may require up to 9 grid zones.

Depending upon the land area one must view, including the various seas, the number of grid zones can range from 6 to 40. Depending upon the resolution required for sufficiently accurate Line-Of-Sight (LOS) or propagation calculations, databases will be on the order of 1 to 10 Gigabytes. These databases can be built into hierarchical nonhomogeneous structures for fast access and minimized storage.

Graphical interfaces with detailed map and entity representations is important from many standpoints. Visualization of complex scenarios is the best approach to verification and validation of simulations and graphics support software. End users can look at the scene and determine quickly if something appears to be operating incorrectly. Once having operated this way, it is difficult to get experienced operators to use facilities that do not provide these insights.

12. SPEED OF SIMULATIONS

Based upon analyses performed by PSI, the distributed battlespace simulation required to support PBA planning turn-around time, accuracy and stability requirements will have to be run at 20 to 100 terahertz rates. That is on the order of 10,000 times faster than today's single processor speeds. If distributed over 100 computers, each would have to have the power of 100 processors. Because of the inefficiencies caused by physical distances, and more importantly going through the layers of communications protocols across multiple operating systems, an innovative parallel processing architecture will have to be brought to bear to meet the speed requirements. This section provides an initial analysis of the factors affecting the requirements for speed of PBA simulations, as well as the difficulties to be overcome to meet these requirements.

REQUIREMENTS FOR SPEED

There are various ways to analyze the requirements for speed to support planning in the AOC. We will start with the current three day (72 hour) cycle of the ATO and define the prediction horizon accordingly as 72 hours. This implies that the overall battlespace simulation would have to cover 72 hours of operation. Running in real time, one simulation would take 72 hours to complete - an unacceptable time period. If one allowed 2 hours to make a single run, then it would have to run 36 times faster than real time. If, after each run, it took 2 hours to review the output, determine what should be changed to improve the likelihood of desired outcomes, and set up the next run, runs could be done every 4 hours, or twice in an 8 hour shift. This would allow for replanning 6 times in a 24 hour period, while looking at the 72 hour time horizon. This is illustrated in Figure 12-1.

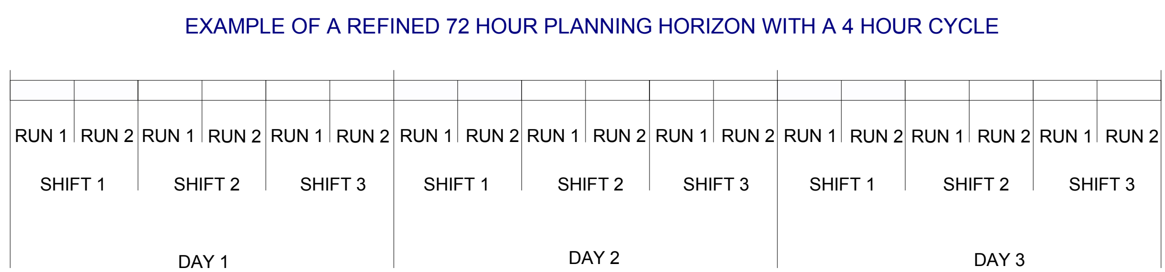

Figure 12-1. Illustration of a short term planning cycle with a long time horizon.

This approach must consider the need for immediate replanning when there are significant situation changes, e.g., when TCT/TST opportunities arise. Intermediate results could be produced, e.g., at 6 hours of simulated time or less. This would allow one to look at the near and intermediate term outcomes at 10 minute intervals, while generating the 72 hour outcomes. Short term changes could be worked independently during each run period, but this would likely require more people and certainly more computer power.

If Monte Carlo type simulations were required to produce a distribution, this may require 20 to 50 runs to estimate the distributions of the end states with sufficient accuracy. If one full planning horizon simulation could be run on a single processor, then multiple simulations could be run simultaneously on parallel processors.

Distributed Processing Considerations

If the battlespace simulation is spread over multiple processors, as shown in Figure 3-5, with feedback loops between processors, then time synchronization must be imposed. Since feedback implies nonlinear behavior, time synchronization becomes critical for validity. If some tools are spread geographically, e.g., for reachback, then there will be significant communication time delays relative to single processor speeds, or even with multiple processors on a local LAN. This leads to the consideration of having copies of the simulations at a single processing site for speed.

This does not preclude the need for subject area experts to provide inputs in replanning cycles. Thus we must allow for getting subject area expert input into a central processing facility if one is to exist. It will also require getting the outputs back. This may be prohibited in a number of cases from a security standpoint.

Best Conditions

With the above in mind, we will consider the speed requirements under best conditions - the entire simulation running in a single computer facility - with no communications latencies. We will then consider how a parallel processing environment may increase speed. We will start by using PSI's JTIDS network management simulation, one of the largest ever built. It has the following properties:

- Runs on a single Intel 3 GHz processor under Windows XP.
- Contains on the order of 125 instances of each platform, each including moving platform dynamics, host traffic models, JTIDS terminals, antenna models, link models that compute propagation path loss in terrain, and instrumentation to measure performance. Each instance is relatively complex.
- Depending upon the message traffic scenario, use of a graphical display, and processor utilization, this simulation runs at approximately real time.

Let's now estimate the multipliers required to run a full-up battlespace simulation.

- To run a 72 hour scenario in 2 hours requires a multiplier of 36X real time.
- A full-up battlespace simulation may require on the order of 10,000 entities. This is a multiplier of 80 (X 125 for JTIDS = 10,000).
- Using 1GHz as the time unit, requires a multiplier of 3 to make up for the speed of the 3 GHz processor used for the JTIDS simulation.

These multipliers = 36 X 80 X 3 = 8640 GHz = 8.64 TeraHertz (THz). We will round this up to 10 THz. It will be at least 10 years before we get to 100 GHz on the best of Moore curves. This will only be $1/100^{th}$ of 10 THz. Three years from now, at 10 GHz, we will need to be about 1000 times faster than the best Intel processor.

Parallel Processing Considerations

Theoretically proving that simulations should run much faster on parallel machines is not nearly as convincing as demonstrating comparative test times from real experiments. To date, the real measures have produced results that are significantly less than anticipated.

There are basically two types of computers that we will consider: (1) multiple processors running under a single operating system; and (2) a tightly coupled cluster of computers (connected via a single/special LAN) each running their own operating system. We note that a cluster may contain type 1multiple processor computers.

Most simulations built for parallel processing today are tailored to the parallel processing environment. As a result, most of these simulations fall into categories that are easily tailored to that environment. These simulations tend to be embarrassingly parallel, or of a particular mathematical form, whereby each processor does a particular piece of the mathematics. An example of an embarrassingly parallel simulation would be doing 100 Monte Carlo runs of a large simulation. Each run can be treated independently and therefore each simulation run can be done on a separate processor. 100 processors can be used very efficiently to do this. The 101^{st} may be much less efficient, unless it simply supervises the others, and does not try to participate in the computation.

Simulations that use a mathematical representation of the system being simulated, whereby they are easily tailored to parallel processors, can also run efficiently. Examples are systems governed by diffusion-like equations, e.g., nuclear, biological, and chemical reactions, whereby each processor can represent the spatial mechanics of a separate part of the space. The PBA problem fits none of these molds. In fact, it contains properties that have been known to thwart efficiency in a parallel processing environment. Our problem calls for running a single simulation on a single processor at 10 THz. Then we must consider the need to run 20 to 50 simulations to produce sufficiently accurate estimates.

Efficiency Considerations

Again, assuming we are running on a single computer facility, there will be inefficiencies that must be accounted for. If in three years we are using 10 GHz processors, then we would need 1000 processors to meet the 10 THz requirement. Processor efficiency using this many processors has been very poor except for the embarrassingly parallel problems. Battlespace simulation does not fit this nor the mathematical mold. Therefore we must take a wag at the processor efficiencies we may expect.

If one achieved 50% using 1000 processors, this would be outstanding by today's measures. This would require processors running at 20 THz. If the efficiency were to be 20%, this would require 40 THz processors. Alternatively, we could be looking at 2000 to 5000 10 GHz processors. We still must consider the need to run 20 to 50 simulations to produce sufficiently accurate estimates.

Summary Of Speed Considerations

What makes the PBA simulation difficult? Some of the properties are listed below.

- The number of totally different entities will be on the order of 100.
- The models of each entity will be totally different.
- The number of instances of many entities will exceed 100 - some may number on the order of 1000.
- Models may have to be run in geographically distributed simulations, viewed by subject area experts.
- Events must be synchronized tightly - in some cases down to milliseconds of simulated time.
- Mathematical models will not support the complex nonlinear rules of entity operation.
- It could take 2000 to 5000 processors running at 10 GHz to produce a single simulation.
- One may need to run 20 to 50 simulations to produce sufficiently accurate estimates.

Approaches To Achieving Speed

From the requirements stated above, any solution to this problem must map into the physical deployment of the individual planning tools and subject area experts that use those tools. This implies an architecture of the type shown in Figures 3-5 and 11-2. Assuming that sufficient processing power can be put at each physical location, one must be concerned with the latencies between computers and particularly the effects of feedback.

Solving the speed problem depends directly on the architecture of the overall battlespace simulation. For example, it may be possible to *flatten* the architecture to remove geographical latencies from the feedback loops. This would remove the need for iterative processing (to gain convergence) that included large geographical latencies.

If the architecture is flattened, then it is likely that Monte Carlo runs may be done in parallel to produce the end state distributions. This will entail an architecture that supports this requirement. Key to any solution is the need to break the battlespace simulation into independent pieces that fit into a flattened architecture. Then, each piece must be able to take advantage of the available hardware that can be used to meet the speed requirements. This implies the ability to automatically generate these simulations so they can run on type 1 parallel processor machines, without having to be retailored for parallel processing.

13. STRATEGIC VERSUS TACTICAL PLANNING

National Level Modeling

When considering strategic level planning, many additional factors come into play. However, the same principles hold. The difference is coming up with the data to test prediction accuracy. Often, the available sources only provide *soft* data. For example, consider the following influential and controllable factors that can affect desired military outcomes.

- Political Environment
- Media Environment
- Financial Environment
- Industrial Environment
- Infrastructure

It is hard to find numbers measuring elements of the first two factors. Some numbers can be derived for the financial and industrial environment, but there are many differences for which numbers are not available. Infrastructure can be described in terms of road networks, communications networks, electric power facilities, etc., where numbers can be used to measure relative differences between nations.

When providing outputs from these types of models, one cannot discard the need to measure accuracy of the predictions of future outcomes. In fact, taking measurements on how well one did when measuring past predictions versus actual outcomes is more important because of the subjective nature of the models and driving force inputs.

Coming up with models of this nature should be similar in approach to coming up with any white box models. One should use all of the information available to produce a model that resembles the actual physical system. The study of how to build these particular models is considered extensively elsewhere, and is beyond the scope of this project.

14. SUMMARY - ACHIEVING PRACTICAL SOLUTIONS

Key properties of the modeling, and simulation environment needed to support practical battlespace planning solutions are provided below:

- **Prediction Accuracy** - Are we able to predict the outcomes of various sorties and missions with sufficient accuracy so that we can pack them tightly? This directly affects speed of the plan as well as surprise, precision/synchronization, and stability.
- **Stability** - Elements of a plan can go awry. Are we able to develop plans that can continue with minimum degradation? Are the resulting plans *robust* in the face of rapid changes?
- **Scalability** –Are we able to incorporate sufficient details in the models to achieve the required prediction accuracy? Are we able to provide an overall simulation architecture and corresponding model architectures to support scalability?
- **Speed** - Is the speed of the planning environment fast enough to react to changes. Can we run sufficient numbers of simulations to produce probability statements with sufficient accuracy in time frames that meet replanning demands?
- **Ease of Use** - Can planners perform replanning functions, using the battlespace simulation environment, with sufficient ease? Can the subject area experts use the interfaces without excessive training?

These properties of the PBA modeling and simulation environment have been addressed to the extent permitted within the scope of this project. One of these topics - speed - has been addressed in other projects. Because of its pertinence to this effort, we reiterated some of these considerations here.

It is most important to understand that, if one makes a probability statement about a future outcome, that statement must be supported by a confidence number. For example, if the probability that A returns safely is > 80%, but the confidence in the statement is less than 50%, what good is the 80% statement? Similarly for stability: if a solution is great, but minor deviations from that solution are likely and can cause bad outcomes, what good is the solution?

There is much work to be done to develop good architectural paradigms for models and simulations of the distributed type required here. Speed must be a major consideration, given that the requirements for prediction accuracy, stability, and scalability must be met. As described in Section 12, some form of parallel processing is going to be required to meet the speed requirements. This must be accounted for in both the model and simulation architectures.

Finally, as in most highly successful technologies, sufficient time is required to take practical advantage of breakthroughs. Given the breakthroughs in model and simulation architecture required by PBA, one must be able to take advantage of order-of-magnitude speed improvements in parallel processing to be of practical use to PBA. We believe that the concepts described in this report form the foundation for those architectures.

15. REFERENCES

[1] *The Theoretical Underpinnings Of Predictive Battlespace.* White Paper submitted to AFRL/IFS, July 3, 2002, Prediction Systems, Inc., Spring Lake, NJ.

[2] Sisti, Alex F., " Dynamic Situation Assessment And Prediction (DSAP)", Air Force Research Laboratory/IFSB, Rome, NY, 2002.

[3] Baumer, Lars, *Identification In Prediction Theory*, IEEE ITW, Kruger National Park, SA, 1999.

[4] Castillo, O and P. Melin, *An Intelligent System For Financial Time Series Prediction Combining Dynamical Systems Theory, Fractal Theory, And Statistical Methods*, IEEE, New York, NY, 1995.

[5] Judd, K. and M. Small, *Towards Long-Term Prediction*, Physica D., Vol. 136, pp 31-34, Elsevier, North Holland, 2000.

[6] Cave, W.C., *Prediction Theory - A Framework For Modeling*, Prediction Systems, Inc., Spring Lake, NJ, 2002.

[7] Cave, W.C., and Busch, T.E., *The Theoretical Underpinnings Of Predictive Battlespace Awareness (PBA)*, Proceedings of SPIE, Vol 5091, Orlando, FL, April 2003.

[8] Gordon, G., *A General Purpose Systems Simulation Program*, Proc. EJCC, Washington, D.C., pp 87-104., MacMillan Publishing Co., New York, 1961.

[9] Gordon, G., **The Application of GPSS V to Discrete System Simulation**, Prentice Hall, Englewood Cliffs, NJ, 1961.

[10] Cave, W.C., *Simulation of Complex Systems*, Prediction Systems, Inc., Spring Lake, NJ, June 2001.

[11] Ramadge, P.J. and W.M. Wonham, "Supervisory Control of a class of discrete-event processes," SIAM J. Control Optimization, vol 25, no.1, pp 206-230, Jan. 1987.

[12] Ramadge, P.J. and W.M. Wonham, "The Control Of Discrete-Event Systems," Proc. IEEE, vol 77, no.1, pp 206-230, Jan. 1989.

[13] Brown, A. and H.C. Yang, "Optimal Worst-Case Circuit Design, " Proceedings NEC, Vol. XXI, 1965.

[14] Cave, W. C., "An Automated Design formulation for Integrated Circuits," Automated Integrated Circuit. Herskowitz, Ed., McGraw Hill, NJ, 1969.

[15] Gelnovatch, V. G. et al, "A 2-4 GHz Integrated Transistor Amplifier Designed by an Optimal Seeking Computer Program," IEEE SSCC Digest, Philadelphia, PA, February, 1970.

[16] Cave, W. C., "Computer Optimization of Microwave Integrated Circuits," IEEE Microwave Integrated Circuits Seminar Proceedings, Monmouth College, NJ, June 1970.

[17] Cave, W.C., "The Constrained Optimal Design System," Proceedings IEEE WESCON, San Francisco, CA, 1971.

[18] Gelnovatch, V. G. et.al., MIMIC Affordability Through Worst Case Design Optimization, Microwave Journal, Vol 31, No 4, MA, April 1988

[19] Schweppe, F., *Uncertain Dynamic Systems*, Prentice Hall, Englewood, NJ, 1973.

[20] Kalman, R., "A New Approach to Linear Filtering and Prediction Problems", Trans. ASME, Series D: Journal Basic Engineering 82, pp. 35-45, 1960.

[21] Mehra, R., "Kalman Filters and Their Applications to Forecasting", TIMS Studies in Management Sciences 12, North-Holland, 1979.

[22] Athans, M. and Falb, P.L., *Optimal Control*, McGraw-Hill, New York, 1966.

[23] Gelb, A., Editor, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.

[24] Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, NY pp. 266, 1965.

[25] *IADS Demonstration Simulation Overview And User's Guide,* Submitted to the EBO Office, AFRL, Rome, NY, Dec. 6, 2001, Prediction Systems, Inc., Spring Lake, NJ.

[26] *Highly Efficient, Scalable Parallel Processors,* Phase I Final Report, DARPA/IPTO SBIR Contract SB022-035 June, 2003, Prediction Systems, Inc., Spring Lake, NJ.

[27] Arthur, W. *Inductive Reasoning and Bounded Rationality (The El Farol Problem)*, Amer. Econ. Review, 84, 406, 1994.

APPENDIX A

Operation DENY Force & CSAR Branch

A Scenario

Developed for

Prediction Systems, Inc.

April 2003

Provided by

Dr. Maris "Buster" McCrabb
DMM Ventures, Inc.

SITUATION

(CAP[1] Stage I): For some time the US monitored reports that Orangeland, a country that traditionally views the US as an adversary, was taking steps that might provide it with weapons of mass destruction (WMD) and the means to deliver them throughout the region and even potentially threaten the US homeland. Orangeland is a "rogue state" in the region and US allies there look to the US for their security. With an increasing amount of tangible evidence of WMD development, especially research into chemical and biological weapons, and the production of ballistic and cruise missiles, the US openly and vigorously protested Orangeland's activities only to be re-buffed with denials. As a precaution, SECDEF directed the theater Combatant Commander, US Mediterranean Command (COMUSMEDCOM) to step up Look, Listen, and Assess operations[2] **(CAP Stage II)**.

As those actions started, chemical attacks occurred at various metropolitan areas within the US. US Northern Command (USNORTHCOM), supported by its AT AOC, assisted with disaster relief operations. SECDEF ordered COMUSMEDCOM to begin contingency planning against Orangeland. The goal is to compel Orangeland to stop WMD or TBM/CM deployment or employment activities. The working title is Operation DENY FORCE. This contingency support on-going, though so far unsuccessful, diplomatic attempts to get Orangeland to the negotiating table.

Input:

SECDEF issues the following guidance via a CJCS WARNING ORDER to COMUSMEDCOM:

- Find, monitor, and assess state of Orangeland's WMD and TBM development;
- Be prepared to strike surgically to defeat development/deployment;
- Minimize the chances of friendly loss and collateral damage during all operations; and
- Consider non-lethal means along with traditional military means for the strike contingency.
- Normal command relationships apply.

[1] Crisis Aaction Planning.

[2] Air Force CONOPS 2020.

Furthermore, the following forces are made available for planning purposes:

- A Global Strike Task Force (GSTF) comprised of F-22, B-2, Global Hawk and Multi-sensor Command & Control Aircraft (MC2A).
- An Air Expeditionary Task Force (AETF).
- The *Stennis* Carrier Battle Group (CVBG) and *Wasp* Amphibious Ready Group (ARG).
- An Interim Brigade Combat Team (IBCT).
- A Block 30-configured Joint Air & Space Operations Center (JAOC)

The WARNORD directs the Commander to provide precise force requirements no later than completion of COA Development (CAP Stage III).

Process:

Upon receipt of the WARNORD, the COMUSMEDCOM assembles the planning team. While in normal operations this could take one of several forms, for the TIE, it is assumed the Commander uses a Standing Joint Force Headquarters (SJFHQ) organizational structure and the Unified Command structure for this theater uses Functional Commands. This scenario concentrates on the JFACC (COMUSMEDAF) and introduces other components only as required. Oftentimes it is the case that the JFC tasks planners to develop Planning Guidance and Commander's Intent as part of the Estimate Process. In this scenario, this step is truncated in order to focus on the SJFHQ and its interaction with the component planning staffs (e.g., the Strategy Division within the JAOC). COMUSMEDCOM issues the following planning guidance:

- Develop strike COA options, with supporting analysis, that:
  - Strike with lethal means only
  - Strike with non-lethal means only
  - Strike with a mix of lethal and non-lethal means
- Wargame each option with attention to attrition and collateral damage
- Provide Branch plans for each COA option that addresses:
  - TCT: Upgraded (e.g., 3rd generation) Orangeland Air Defenses (e.g., SA-10 surface-to-air missiles, SU-37 Flanker aircraft), or any TBM/CM (cruise missile) deployment/employment activities.
  - CSAR

COMUSMEDCOM also issued the following Commander's Intent:

End State

- No WMD/TBM threat to region/US
- Desired effect: deter deployment of WMD/TBM; on order, disrupt (destroy if ordered) WMD/TBM development/deployment
- Purpose: regional stability & US security
- Method: surgical strike
- Risk: low to US forces; medium for collateral damage

For this scenario, a modified planning process is used for CAP Stage III. The basic process comes from Joint Pub (JP) 5-00.1. The key modifications include specifying mechanism between actions (strategy) and results (objectives), and specifying campaign (operational) assessment as separate from, but building upon, combat assessment. Two sub-processes are key. The first is IPB. This sub-process is expanded upon and re-named Operational Net Assessment (ONA) in the JFCOM/J9 SJFHQ *Concept of Employment*. ONA shares JIPB in common with the Air Force's Predictive Battlespace Awareness (PBA) concept. PBA combines IPB (with obvious emphasis on the air & space aspects of IPB) with ISR planning and management. ONA builds upon IPB (with obvious emphasis on the Joint aspects of IPB) by offering an analysis of what actions might be taken to achieve JFC desired effects.

The second sub-process is a modification of the effects-based planning framework offered by then-Colonel David A. Deptula in *Firing For Effect*. The small change is making explicit the interactive nature of IPB (identifying targets) and effects-based planning (identifying specific effects).

As the planning teams assemble via the SJFHQ-directed collaborative network, the planning tasks are broken out through the staff estimate processes. The input is commander's intent. The output is a COA in the form of an ETO (Effects Tasking Order). In between, options are generated and wargamed. The results are presented to the JFC for decision. These processes and products are highly interactive between the SJFHQ and component planning staffs. The JFACC, for example, would direct preparation of the air estimate of the situation.

Within the SJFHQ, several teams, cells, boards and agencies are involved. For example, ONA teams start with center-of-gravity (COG) analysis in support of determining Orangeland's PMESII (political, military, economic, social informational, and infrastructure) strengths and vulnerabilities. EBO teams begin mission analysis to determine specified, implied, and essential tasks in the commander's intent. Logistics teams begin with forces made available for planning to start the detailed time phased force and deployment data processes.

Planners develop strategy options based on three mechanisms. Each is tied directly to the political-military leadership model of Orangeland. Analysts determined that model showed an autocratic, oligarchic leadership with limited points of access. The first mechanism is destruction, that is, a classic attrition, or brute force, strategy. The defeat mechanism[3] is that by destroying Orangeland's WMD or TBM/CM capabilities, they could no longer threaten their regional neighbors or the US homeland with those systems.

The second mechanism is disruption, that is a denial strategy. The defeat mechanism is that by disrupting Orangeland's capability to deploy or employ those systems, it is prevented from threatening others, at least for the moment, and thus, by being denied the means of obtaining their objective (presumed to be regional hegemony) Orangeland would be more likely to negotiate.

The final strategy option is coercion through holding these WMD and TBM/CM systems at risk. The defeat mechanism is that the costs of losing these systems outweigh the benefit of having them so that if Orangeland believes the US could indeed destroy these systems (see the first mechanism); they would more likely negotiate then risk losing these capabilities.

These three options meld into specific COA options by varying the weights, or probabilities, of the three mechanisms. For example, COA option one is heavily weighted towards attriting Orangeland's WMD and TBM/CM capabilities through direct attack on these systems and against a select set of targets within these systems' value chains. Thus, destruction combines with disruption with at least the implicit assumption that, over the course of the campaign, Orangeland will be coerced into negotiation rather than see the continual loss of these valuable—and expensive—systems.

These COA options are wargamed against specified criteria.[4] One is the likelihood of US actions provoking Orangeland employment of these capabilities under fear of a "use or lose" scenario. US planners keenly recall charges that Operation ALLIED FORCE in 1999 provoked Yugoslavian leader Slobodon Milosevic to accelerate and intensify the ethic cleansing actions against the Albanian Kosovors. COA option one scores low on this criterion.

---

[3] A defeat, or "overcoming," mechanism is normally only specified as such at the national and theater strategic levels (using *Universal Joint Task List* nomenclature). It is defined as the explanation on how the obstacle preventing attainment of a goal is overcome. In a zero-sum situation, it can be viewed as "winning" versus "losing." However, in on-zero sum situations, like a Humanitarian Relief Operations (HUMRO) or deterrence operations where the strategic aim is a negative (such as peace keeping) defeat mechanisms still exist even though terms like winning or losing are not normally used except as propaganda (e.g., "we're winning the peace"). At the theater operational and below levels, it is normally simply referred to as mechanism—the explanation on why the direct or indirect action causes the effect (or outcome).

[4] Each COA option is assumed to be adequate, acceptable, and complete. The evaluation of the COA options against those criteria is omitted.

Option two, that disrupts Orangeland's WMD and TBM/CM capability not by directly attacking these systems but rather through lethal and non-lethal attacks against supporting systems and infrastructure, scores higher in wargames on this "propensity to provoke" criteria. On the other hand, option two scores lower on three additional criteria. One is time required to accomplish the desired effect. Since Orangeland is best able to determine whether their WMD or TBM/CM capabilities are "disrupted" to such an extent they feel compelled to negotiate, the likelihood is these operations will take longer than under option one. This length is exacerbated by the belief that non-lethal effects take longer to be felt than effects instigated by direct physical actions. The other two criteria scores derive from the "time-to-complete" criteria. These are "probability of friendly losses" and "probability of collateral damage." The more actions taken, the more likely something goes wrong. These likelihood's are mitigated somewhat by the non-lethal force applications that form part of option two (and three) but not option one.

Option 3 scores better than the other two on the collateral damage, friendly loss and provocation criteria. It scores worst on time-to-complete and likelihood of attaining the desired effect criteria. That is because it emphasizes non-lethal attacks that demonstrate US capability to harm Orangeland's WMD and TBM/CM capabilities and support systems without actually damaging any. Hence, the success of this option hinges critically on the credibility of the threat. That credibility is a function of resolve to persevere and the willingness to escalate to more lethal means if Orangeland refuses to negotiate.

Across all the criteria, option two scores highest. This is the one recommended to the commander by the planners. The JFC, based upon consultations with SECDEF and others, decides the coercive nature of the attacks is best served by attacking up the WMD/TBM "value chain." This accomplishes three things: demonstrates US ability of hold these assets at risk; demonstrates US resolve to do so; and should not force Orangeland into a "use or lose" situation by directly attacking these weapon systems themselves unless Orangeland shows signs of preparing to deploy or employ any of these weapons

The JFC orders that branches be fully fleshed out. The most critical is the provocation scenario. The commander wants ISR assets planned and managed with an eye towards detecting any Orangeland activity that might indicate WMD or TBM/CM deployment or employment. Further, the JFC directs contingency strike planning that can respond quickly to any indications or warnings found by ISR. The second contingency is the loss of an aircraft. Besides the obvious desire to recover any lost pilot, the JFC fears the pilot may become a bargaining advantage Orangeland could exploit to mitigate the US actions.

Output:

The COA sent to the components for detailed planning has these elements:

1. What (desired effect): Disrupt (destroy if ordered) Orangeland's WMD and TBM/CM capability to deploy or employ.
2. How (strategy): Attack Orangeland's supporting systems, such as WMD/TBM R&D facilities, and infrastructure elements, such as Command and Control (C2) or storage, that directly influence their WMD or TBM/CM deployment or employment capabilities.
3. Why (rationale): US desire regional stability and reduced threat to the US homeland. Orangeland WMD and TBM/CM capabilities, and their apparent willingness to use them, threaten both these objectives. Therefore reduction in this threat increases the probability the US attains its objectives.
4. Why (mechanism): If Orangeland's WMD and TBM/CM capabilities are disrupted, then they will more likely negotiate with the US because Orangeland would fear further US action to directly destroy these valuable, and expensive to replace, capabilities.
5. With (resources): Forces made available for planning.
6. Where (location): To be determined by component planners.
7. When (time): Within 24 hours of receipt of EXECUTE ORDER.
8. Who (units): Initial strike by the Global Strike Task Force with follow-on (if required) strike from the CVBG and AETF. ARG and IBCT on call.

ROE: take all steps to minimize the likelihood of friendly loss and collateral damage; maximize the use of non-lethal means where appropriate.

CSAR Branch

CSAR is somewhat different from a TCT (time critical target) problem. The former is more sensitive to location, made worse by CCD (cover, concealment, and deception) efforts, while generally less sensitive to time than a TCT. On the other hand, a TCT is less sensitive to location--within the ellipse of planned weapons such as AGM-130 is generally good enough--but the timing window can be less than 10 minutes from “find” to “engage.” Second, CSAR assets are dedicated assets whereas TCT assets may have to be diverted from other missions. Thus, TCT decisions require more trade-off analysis, under stricter time constraints, than CSAR.

The one mitigating factor, of course, is that if all goes as planned, the friendly survivor is an active participant in the rescue attempt whereas in classic CCD, the adversary is taking measures, both active and passive, to thwart detection. On the other hand, in a CSAR scenario the survivor is trying to thwart the adversary detecting their location. Further, CSAR planners must account for the situation where, for a myriad of reasons, the survivor’s electronic apparatus is not working or, perhaps due to physical incapacitation, the survivor is completely passive in their rescue.

DAR (Designated Areas for Recovery) data is generated as part of planning. Also supporting is Enemy Order of Battle (EOB) data so planners understand where the highest threats are likely to be encountered. Obviously, that is where the SEAD (Suppression of Enemy Air Defenses) assets are concentrated but it is also, where the greatest likelihood of a shoot down occurs.

When the shoot down occurs (that is, CSAR execution), ISR assets are re-configured to support the rescue. Of great importance is the GMTI data that monitors vehicular traffic in and around the survivor’s location. This is an input into the decision-making process on how much time is available to prevent the survivor’s capture.